

用户手册

78K0S/KB1+

8 位单片微控制器

μPD78F9232
μPD78F9234

版次： U17446EJ2V0UD00 (第一版)
印次： 2005 年 4 月 NS CP(K)

NEC 公司 2005
日本印制

[备忘录]

CMOS 设备注意事项

① 输入引脚处的电压适用波形

因输入噪声或反射波产生的干扰可能产生故障。如果由于噪声等影响，CMOS 设备的输入端电平停留在 VIL(最大)和 VIH(最小)之间，该设备可能产生故障。当输入电平固定时，和在传输阶段当输入电平在 VIL(最大)和 VIH(最小)之间时，切记避免波动噪声进入设备。

② 未使用的 CMOS 输入引脚的处理

CMOS 设备输入端未连接将会产生工作故障。如果输入引脚未连接，则由于噪音等原因可能会产生内部输入电平，从而导致故障。CMOS 设备的运行与 Bipolar 或 NMOS 设备不同。必须借助上拉或下拉电路使 CMOS 设备的输入电平固定在高电平或是低电平。如果每个未用的引脚可作为输出引脚的话，那么它们可以通过电阻与 VDD 或 GND 连接。对未使用引脚的处理因设备而异，必须遵循与设备相关的规定和说明。

③ 半导体的 ESD 防护措施

如果 MOS 设备周围有强电场，将会击穿氧化栅极，从而影响设备的运行。因此必须采取措施，尽可能防止静电产生，而且一旦有静电，必须立即释放。对于环境必须有适当的控制。如果空气干燥，应当使用加湿器。建议避免使用绝缘体，因为它们容易产生静电。半导体设备的存放和运输必须在抗静电容器、抗静电屏蔽袋或导电材料容器中进行。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手套以保证良好接地，不能用手直接接触半导体设备。对于装配有半导体设备的 PW 板也应采取类似的静电防范措施。

④ MOS 设备初始化之前的状态

上电并不能确定 MOS 设备的初始状态。MOS 设备的生产工艺也没有确定其初始操作状态。在刚刚上电之后，具有复位功能的 MOS 设备并没有被初始化，因此上电不能保证输出引脚的电平，I/O 设置和寄存器的内容。设备在收 到复位信号后才进行初始化，具有复位功能的设备在上电后必须立即进行复位操作。

⑤ 上/下电顺序

在器件内部操作和外部接口使用不同供电电源的情况下，上电顺序的规则是在内部电源打开后再打开外部供电电源。下电的规则是先关掉外部供电电源，然后是内部电源。使用如上相反的顺序上下电可能导致器件内部元件过压，从而导致故障和因为不正常的电流致使内部元件退化。

正确的上下电顺序因设备而异，必须遵循与设备相关的规定和说明。

⑥ 下电状态的信号输入

在器件没上电时，不要输入信号或者一个 I/O 上拉电源。因输入该信号或 I/O 上拉电源产生的电流注射可能产生故障和产生流过器件的不寻常的电流，从而在那时产生内部元件退化。在下电状态下的信号输入因设备而异，必须遵循与设备相关的规定和说明。

注意事项：该产品使用 SST 公司授权的 SuperFlash® 技术。

区域信息

本文档中的某些信息可能因国家不同而有所差异。用户在使用任何一种 NEC 产品之前，请与当地的 NEC 办事处联系，以获取权威的代理商和发行商信息。请验证以下内容：

设备的可用性

定货信息

产品发布进度表

相关技术资料的可用性

开发环境要求（例如：要求第三方工具和组件，主计算机，电源插头，AC 供电电源等）

网络要求

此外，对于商标、注册商标、出口限制条款和其他法律规定，不同的国家也有不同的要求。

【全球支持】

http://www.necel.com/en/support/support.html

NEC Electronics America,Inc.(U.S.)
Santa Clara,Callfomia
Tel:408-588-6000
800-366-9786

NEC Electronics(Europe)GmbH
Duesseldorf,Germany
Tel:0211-65030

- Sucursal en Espana
  Madrid,Spain
  Tel:091-504-2787

- Succursale Francaise
  Velizy-Villacoublay,France
  Tel:01-30-675800

- Filiale Italiana
  Milano,Italy
  Tel:02-667541

- Branch The Netherlands
  Eindhoven,TheNetherlands
  Tel:040-2445845

- Tyskland Filial
  Taeby,Sweden
  Tel:08-6380820

- United Kingdom Branch
  Milton Keynes,UK
  Tel:01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kongf
Tel:2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul,Korea
Tel:02-558-3737

NEC Electronics Shanghai,Ltd.
ShangHai,P.R.China
Tel:021-6888-5400

NEC Electronics Taiwan Ltd.
Taipei,Taiwan
Tel:02-2719-2377

NEC Electronics Singapore Pte.Led.
Novena Square,Singapore
Tel:6253-8311

引言

读者对象

本手册适用于那些希望了解 78K0S/KB1+ 产品功能，并设计开发相关应用系统和程序的用户。

主要产品如下。

- 78K0S/KB1+： μPD78F9232，78F9234

目的

本手册用于帮助用户了解下面**组织**中描述的功能。

组织

78K0S/KB1+手册主要分为两个部分: 本手册和指令手册 (通用 78K/0S 系列)。

78K0S/KB1+ 用户手册	78K/0S 系列 指令用户手册
• 引脚功能 • 内部模块功能 • 中断 • 其他内置外设功能 • 电气特性（目标值）	• CPU 功能 • 指令集 • 指令描述

手册使用方法

在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

◊ 如何理解 78K0S/KB1+的整体功能
→ 按**目录**顺序阅读本手册。
◊ 如何理解寄存器格式
→ 括号中的二进制位名在 RA78K0S 中被定义为保留字, 并且在 CC78K0S 中用 #pragma sfr 伪指令定义为一个 sfr 变量。
◊ 如何获悉某寄存器的详细信息
→ 可参阅**附录 C 寄存器索引**。
◊ 如何获悉 78K/0S 系列指令的详细信息
→ 可参阅 **78K/0S 系列指令用户手册** (U11047E)。
◊ 如何获悉 78K0S/KB1+ 的电气特性
→ 可参阅**第二十一章 电气特性 (目标值)**。

规定 数据规则: 数据的高位部分在左边，低位部分在右边

低电平有效表示法: $\overline{\times\times\times}$（在引脚和信号名称上划一条线）

注: 文中用注标注的相关术语的脚注

注意事项: 需要特别关注的信息

备注: 补充信息

数的表示法: 二进制 ... ×××× 或 ××××B

十进制 ... ××××

十六进制 ... ××××H

相关文档 本手册中指出的相关文档包括了最初的版本，但未注明。

相关设备文档

文档名称	文档编号
78K0S/KB1+用户手册	本手册
78K/0S 系列指令用户手册	U11047E

开发工具(软件)文档 (用户手册)

文档名称		文档编号
RA78K0S 汇编包	操作	U16656E
	语言	U14877E
	结构化汇编语言	U11623E
CC78K0S C 编译器	操作	U16654E
	语言	U14872E
ID78K0S-NS 集成调试器 2.52 版	操作	U16584E
ID78K0S-QB 集成调试器 2.81 版	操作	U17287E
PM＋ 5.20 版		U16934E

开发工具（硬件）文档（用户手册）

文档名称	文档编号
IE-78K0S-NS 在线仿真器	U13549E
IE-78K0S-NS-A 在线仿真器	U15207E
QB-78K0SKX1MINI 在线仿真器	U17272E

注意事项 以上列出的相关文档在无任何声明条件下可能会作修改。读者应使用每个文档的最新版本用于设计。

Flash 存储器编程相关文档

文档名称	文档编号
PG-FP4 Flash 存储器编程用户手册	U15260E
PG-FPL2 Flash 存储器编程用户手册	U17307E

其他相关文档

文档名称	文档编号
半导体选择指南 - 产品和封装	X13769X
半导体设备安装手册	注
NEC 半导体设备质量等级	C11531E
NEC 半导体设备可靠性/质量控制系统	C10983E
半导体设备防静电 ESD 指南	C11892E

注　可参阅“半导体设备安装手册”网站（http://www.necel.com/pkg/en/mount/index.html）。

注意事项　以上列出的相关文档在无任何声明条件下可能会作修改。读者应使用每个文档的最新版本用于设计。

目录

第一章　　概述

1.1　特征

- 指令最短执行时间可以在高速(0.2μs)和低速(3.2μs)之间改变(CPU 时钟为 10 MHz)
- 通用寄存器：8 位 × 8 个寄存器
- ROM 和 RAM 的容量

产品编号＼项目	程序存储器 (Flash 存储器)	存储器(内部高速 RAM)
μPD78F9232	4 KB	256 字节
μPD78F9234	8 KB	256 字节

- 内置上电清零(POC)电路和低电压检测器 (LVI)
- 内置看门狗定时器(基于内部低速 Ring-OSC 时钟工作)
- I/O 端口：26
- 定时器：4 通道
  - 16 位定时器/事件计数器：　1 通道
  - 8 位定时器：　2 通道
  - 看门狗定时器：　1 通道
- 串行接口： UART (支持 LIN (本地互联网络) 总线) 1 通道
- 片内乘法器： 8 位 x 8 位= 16 位
- 10 位分辨率 A/D 转换器：4 通道
- 供电电压： VDD = 2.0～5.5 V[注]
- 工作环境温度： TA = −40 ～ +85°C

注　因为上电清零(POC)的检测电压 (V_{POC})在 2.1 V ± 0.1 V 之间，所以本产品的电压范围是 2.2 ～ 5.5 V。

1.2　应用领域

- 汽车电子
  - 车身电子系统控制（例如电动门窗，门锁控制等）
  - 控制系统的次级微控制器
- 家用电器
  - 电动牙刷
  - 电动剃须刀
- 玩具
- 工业设备
  - 传感器和开关控制
  - 电源工具

1.3　订购信息

产品编号	封装	产品等级
μPD78F9232MC(T)-5A4	30 引脚塑封 SSOP	标准
μPD78F9232MC(S)-5A4	30 引脚塑封 SSOP	标准
μPD78F9232MC(R)-5A4	30 引脚塑封 SSOP	标准
μPD78F9232MC(T)-5A4-A	30 引脚塑封 SSOP	标准
μPD78F9232MC(S)-5A4-A	30 引脚塑封 SSOP	标准
μPD78F9232MC(R)-5A4-A	30 引脚塑封 SSOP	标准
μPD78F9234MC(T)-5A4	30 引脚塑封 SSOP	标准
μPD78F9234MC(S)-5A4	30 引脚塑封 SSOP	标准
μPD78F9234MC(R)-5A4	30 引脚塑封 SSOP	标准
μPD78F9234MC(T)-5A4-A	30 引脚塑封 SSOP	标准
μPD78F9234MC(S)-5A4-A	30 引脚塑封 SSOP	标准
μPD78F9234MC(R)-5A4-A	30 引脚塑封 SSOP	标准

备注　μPD78F9232MC(xx)- 5A4-A 和 78F9234MC(xx)-5A4-A 是无铅产品。
xx ：T，S，R

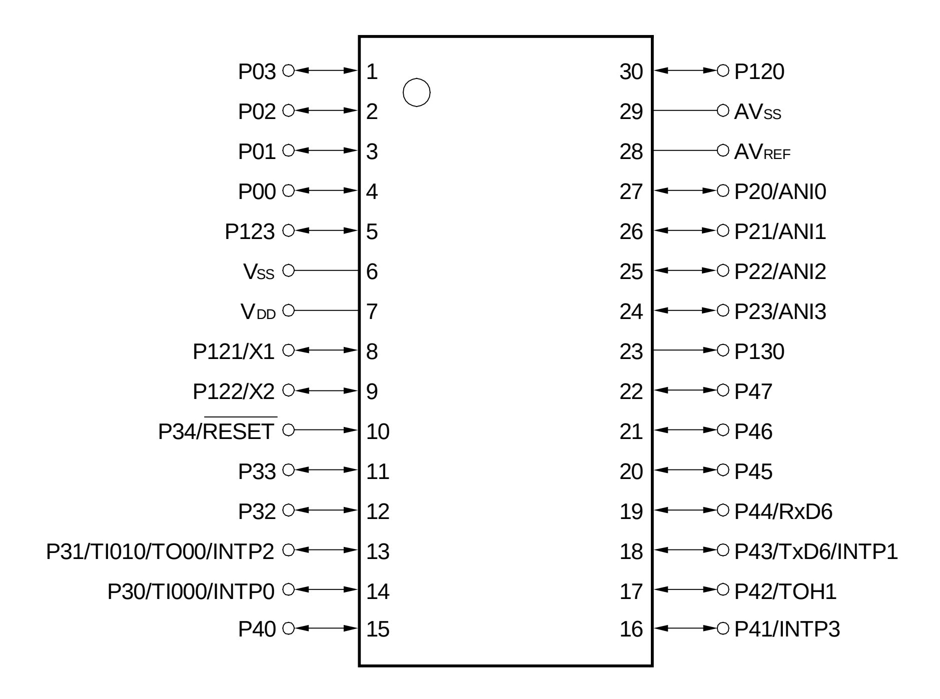
请参考NEC公司发布的“NEC半导体设备质量等级”（文档编号 C11531E），以了解设备的质量等级规范和推荐应用。

以下是 78K0S/KB1+ 标准等级产品更进一步的分类：

(T)：　普通生产管理
(S)：　基于单独合同的生产管理
(R)：　基于汽车电子类的生产管理

1.4　引脚配置(俯视图)

30 引脚塑封 SSOP



注意事项　将 AV_{SS} 引脚和 V_{SS} 引脚连接在一起。

ANI0～ANI3：	模拟输入	P130：	端口 13
AV_{REF}：	模拟参考电压	$\overline{RESET}$：	复位
AV_{SS}：	模拟地	RxD6：	接收数据
INTP0～INTP3：	外部中断输入	TI000，TI010：	定时器输入
P00～P03：	端口 0	TO00，TOH1：	定时器输出
P20～P23：	端口 2	TxD6：	发送数据
P30～P34：	端口 3	V_{DD}：	电源
P40～P47：	端口 4	V_{SS}：	地
P120～P123：	端口 12	X1，X2：	晶体振荡器 (X1　输入时钟)

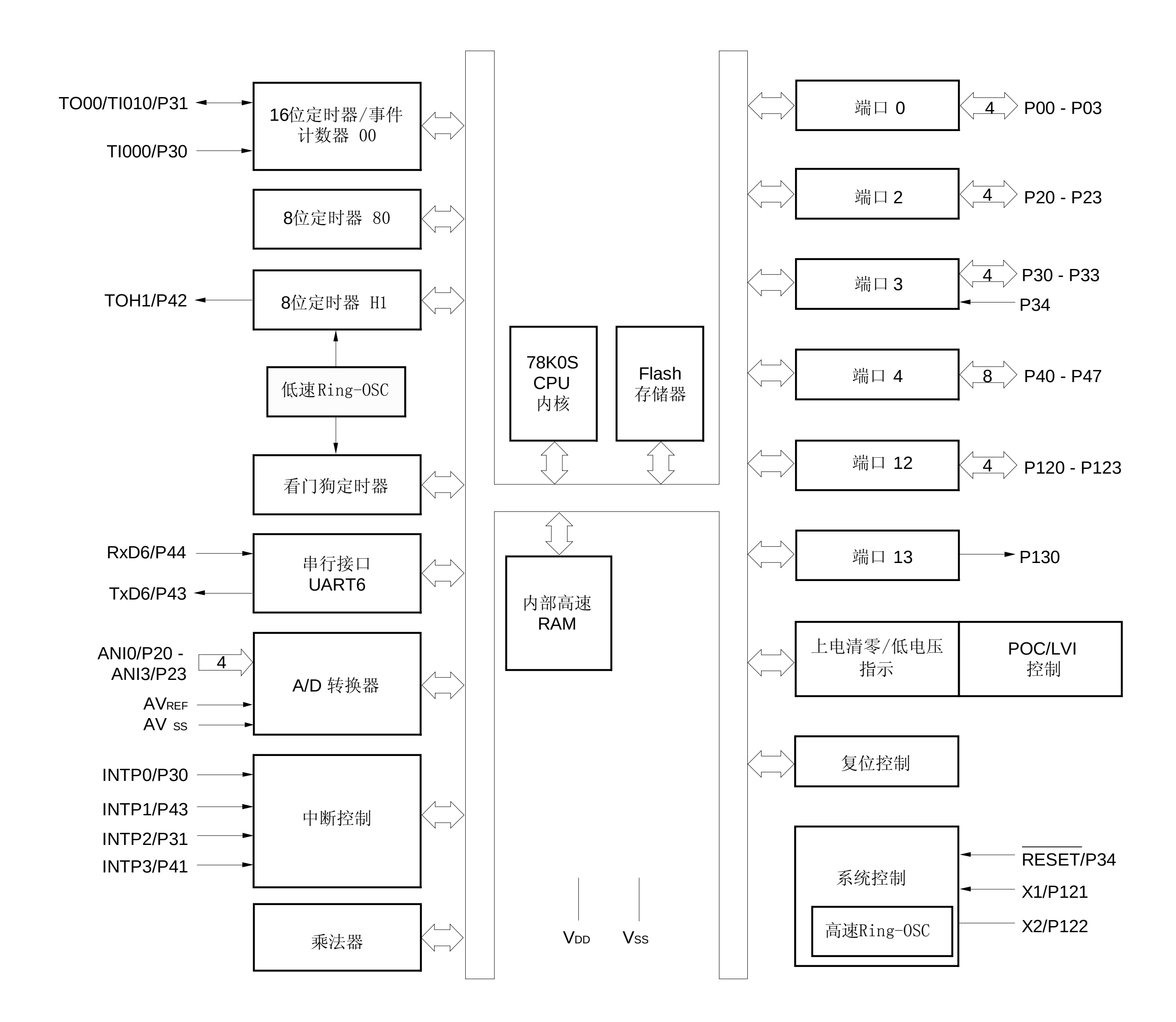
1.5　78K0S/Kx1+系列介绍

下面的表格给出了78K0S/Kx1+系列产品。

<table>
<tr><td colspan="2">产品编号
项目</td><td colspan="2">78K0S/KY1+</td><td colspan="2">78K0S/KA1+</td><td>78K0S/KB1+</td></tr>
<tr><td colspan="2">引脚数目</td><td colspan="2">16 引脚</td><td colspan="2">20 引脚</td><td>30 引脚</td></tr>
<tr><td rowspan="2">内部存储器</td><td>Flash 存储器</td><td colspan="2">1 KB，2 KB，4 KB</td><td>2 KB</td><td>4 KB</td><td>4 KB，8 KB</td></tr>
<tr><td>RAM</td><td colspan="2">128 字节</td><td>128 字节</td><td>256 字节</td><td>256 字节</td></tr>
<tr><td colspan="2">电源电压</td><td colspan="5">VDD = 2.0～5.5 V 注</td></tr>
<tr><td colspan="2">指令最短执行时间</td><td colspan="5">0.20 μs（10 MHz， VDD = 4.0～5.5 V）
0.33 μs（6 MHz， VDD = 3.0～ 5.5 V）
0.40 μs（5 MHz， VDD = 2.7～5.5 V）
1.0 μs（2 MHz， VDD = 2.0～5.5 V）</td></tr>
<tr><td colspan="2">系统时钟（振荡器频率）</td><td colspan="5">内部高速 Ring-OSC 振荡器（8 MHz（典型值））
晶体/陶瓷振荡器（1～10 MHz）
外部时钟输入振荡器（1～10 MHz）</td></tr>
<tr><td colspan="2">TMH1 和 WDT 时钟（振荡器频率）</td><td colspan="5">内部低速 Ring-OSC 振荡器（240 kHz（典型值））</td></tr>
<tr><td rowspan="3">端口</td><td>CMOS I/O</td><td colspan="2">13</td><td colspan="2">15</td><td>24</td></tr>
<tr><td>CMOS 输入</td><td colspan="2">1</td><td colspan="2">1</td><td>1</td></tr>
<tr><td>CMOS 输出</td><td colspan="2">–</td><td colspan="2">1</td><td>1</td></tr>
<tr><td rowspan="4">定时器</td><td>16 位（TM0）</td><td colspan="5">1 通道</td></tr>
<tr><td>8 位（TMH）</td><td colspan="5">1 通道</td></tr>
<tr><td>8 位（TM8）</td><td colspan="2">–</td><td colspan="3">1 通道</td></tr>
<tr><td>WDT</td><td colspan="5">1 通道</td></tr>
<tr><td colspan="2">串行接口</td><td colspan="2">–</td><td colspan="3">支持 LIN 总线的 UART： 1 通道</td></tr>
<tr><td colspan="2">A/D 转换器</td><td colspan="5">10 位： 4 通道(2.7 ～ 5.5V)</td></tr>
<tr><td colspan="2">乘法器（8 位 × 8 位）</td><td colspan="4">–</td><td>提供</td></tr>
<tr><td rowspan="2">中断</td><td>外部</td><td colspan="2">2</td><td colspan="3">4</td></tr>
<tr><td>内部</td><td colspan="2">5</td><td colspan="3">9</td></tr>
<tr><td rowspan="4">复位</td><td>$\overline{\text{RESET}}$ 引脚</td><td colspan="5">提供</td></tr>
<tr><td>POC</td><td colspan="5">2.1 V ± 0.1 V</td></tr>
<tr><td>LVI</td><td colspan="5">提供（软件选择）</td></tr>
<tr><td>WDT</td><td colspan="5">提供</td></tr>
<tr><td colspan="2">工作温度环境</td><td colspan="5">–40 ～ +85°C</td></tr>
</table>

注　因为上电清零(POC)的检测电压（V_{POC})在2.1 V ± 0.1 V之间，所以本产品的电压范围是2.2 ～ 5.5 V。

1.6 框图



TO00/TI010/P31
TI000/P30
16位定时器/事件计数器 00
8位定时器 80
TOH1/P42
8位定时器 H1
低速Ring-OSC
看门狗定时器
RxD6/P44
TxD6/P43
串行接口 UART6
ANI0/P20 - ANI3/P23
4
AVREF
AVSS
A/D 转换器
INTP0/P30
INTP1/P43
INTP2/P31
INTP3/P41
中断控制
乘法器
78K0S CPU 内核
Flash 存储器
内部高速 RAM
VDD
VSS
端口 0
4
P00 - P03
端口 2
4
P20 - P23
端口 3
4
P30 - P33
P34
端口 4
8
P40 - P47
端口 12
4
P120 - P123
端口 13
P130
上电清零/低电压指示
POC/LVI 控制
复位控制
系统控制
高速Ring-OSC
RESET/P34
X1/P121
X2/P122

1.7 功能概述

项目			μPD78F9232	μPD78F9234
内部存储器	Flash 存储器		4 KB	8 KB
	高速 RAM		256 字节	
存储器空间			64 KB	
X1 输入时钟(振荡器频率)			晶体/陶瓷/外部时钟输入： 10 MHz (VDD = 2.0 ～ 5.5 V)	
Ring-OSC 时钟	高速(振荡器频率)		内部 Ring-OSC： 8 MHz (典型值)	
	低速(TMH1 和 WDT)		内部 Ring-OSC： 240 kHz (典型值)	
通用寄存器			8 位 × 8 个寄存器	
指令最短执行时间			0.2 μs/0.4 μs/0.8 μs/1.6 μs/3.2 μs (X1 输入时钟：f_X = 10 MHz)	
乘法器			8 位 x 8 位 = 16 位	
I/O 端口			总计： 26 引脚 CMOS I/O： 24 引脚 CMOS 输入： 1 引脚 CMOS 输出： 1 引脚	
定时器			• 16 位定时器/事件计数器： 1 通道 • 8 位定时器 (定时器 H1)： 1 通道 • 8 位定时器 (定时器 80)： 1 通道 • 看门狗定时器： 1 通道	
		定时器输出	2 引脚 (PWM： 1 引脚)	
A/D 转换器			10 位分辨率 × 4 通道	
串行接口			支持 LIN 总线的 UART 模式： 1 通道	
向量中断源		外部	4	
		内部	9	
复位			•使用 $\overline{\text{RESET}}$ 引脚复位 •使用看门狗计时器进行内部复位 •使用 POC 进行内部复位 •使用低电压检测器进行内部复位	
电源电压			VDD = 2.0 ～ 5.5 V[注]	
工作环境温度			TA = −40 ～ +85℃	
封装形式			30 引脚塑封 SSOP	

注 因为上电清零(POC)的检测电压 (V_{POC})在 2.1 V ± 0.1 V 之间，所以本产品的电压范围是 2.2 ～ 5.5 V。

第二章　引脚功能

2.1　引脚功能列表

(1)　端口引脚

<table>
<tr><th>引脚名称</th><th>I/O</th><th colspan="2">功能</th><th>复位后</th><th>复用功能引脚</th></tr>
<tr><td>P00 ～ P03</td><td>I/O</td><td colspan="2">端口 0
4 位 I/O 端口
可以位选输入/输出方式
可以通过软件设置连接内部上拉电阻</td><td>输入</td><td>–</td></tr>
<tr><td>P20 ～ P23</td><td>I/O</td><td colspan="2">端口 2
4 位 I/O 端口
可以位选输入/输出方式
可以通过软件设置连接内部上拉电阻</td><td>输入</td><td>ANI0 ～ ANI3</td></tr>
<tr><td>P30</td><td rowspan="4">I/O</td><td rowspan="5">端口 3</td><td rowspan="4">可以位选输入/输出方式
可以通过软件设置连接内部上拉电阻</td><td rowspan="4">输入</td><td>TI000/INTP0</td></tr>
<tr><td>P31</td><td>TI010/TO00/
INTP2</td></tr>
<tr><td>P32</td><td>–</td></tr>
<tr><td>P33</td><td>–</td></tr>
<tr><td>P34[注]</td><td>输入</td><td>仅输入</td><td>输入</td><td>$\overline{RESET}$[注]</td></tr>
<tr><td>P40</td><td rowspan="8">I/O</td><td colspan="2" rowspan="8">端口 4
8 位 I/O 端口
可以位选输入/输出方式
可以通过软件设置连接内部上拉电阻</td><td rowspan="8">输入</td><td>–</td></tr>
<tr><td>P41</td><td>INTP3</td></tr>
<tr><td>P42</td><td>TOH1</td></tr>
<tr><td>P43</td><td>TxD6/INTP1</td></tr>
<tr><td>P44</td><td>RxD6</td></tr>
<tr><td>P45</td><td>–</td></tr>
<tr><td>P46</td><td>–</td></tr>
<tr><td>P47</td><td>–</td></tr>
<tr><td>P120</td><td rowspan="4">I/O</td><td colspan="2" rowspan="4">端口 12
4 位 I/O 端口
可以位选输入/输出方式
仅 P120 和 P123 能通过软件设置连接内部上拉电阻</td><td rowspan="4">输入</td><td>–</td></tr>
<tr><td>P121[注]</td><td>X1[注]</td></tr>
<tr><td>P122[注]</td><td>X2[注]</td></tr>
<tr><td>P123</td><td>–</td></tr>
<tr><td>P130</td><td>输出</td><td colspan="2">端口 13
1 位输出端口</td><td>输出</td><td>–</td></tr>
</table>

注　复用功能的详细设置，请参见 第十八章　可选字节。

注意事项　　P121/X1 和 P122/X2 引脚复位期间为下拉状态。

(2) 非端口引脚

引脚名称	I/O	功能	复位后	复用功能引脚
INTP0	输入	当指定的有效沿（上升沿、下降沿或双沿）到来时，产生外部中断请求	输入	P30/TI000
INTP1				P43/TxD6
INTP2				P31/TI010/TO00
INTP3				P41
RxD6	输入	异步串行接口的串行数据输入	输入	P44
TxD6	输出	异步串行接口的串行数据输出	输入	P43/INTP1
TI000	输入	输入到16位定时器/事件计数器00的外部计数时钟 输入到16位定时器/事件计数器00的捕捉寄存器（CR000和CR010）的捕捉触发	输入	P30/INTP0
TI010		输入到16位定时器/事件计数器00的捕捉寄存器（CR000）的捕捉触发		P31/TO00/INTP2
TO00	输出	16位定时器/事件计数器00输出	输入	P31/TI010/INTP2
TOH1	输出	8位定时器H1输出	输入	P42
ANI0 ～ ANI3	输入	A/D 转换器的模拟输入	输入	P20 ～ P23
AVREF	–	A/D转换器的参考电压	–	–
AVSS	–	A/D转换器的地，与VSS 的电位相同	–	–
$\overline{\text{RESET}}$ 注	输入	系统复位输入	–	P34 注
X1 注	输入	系统时钟振荡的晶体/陶瓷振荡器的连接 外部时钟输入	–	P121 注
X2 注	–	系统时钟振荡的晶体/陶瓷振荡器的连接	–	P122 注
VDD	–	电源电压	–	–
VSS	–	地	–	–

注 复用功能的详细设置，请参见 第十八章 可选字节。

注意事项 P121/X1和P122/X2引脚复位期间为下拉状态。

2.2 引脚功能描述

2.2.1 P00 ～ P03（端口 0）

端口 0 是一个 4 位 I/O 端口，可通过端口模式寄存器 0（PM0）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 0（PU0）设置为使用内部上拉电阻。

2.2.2 P20 ～ P23（端口 2）

端口 2 是一个 4 位 I/O 端口。除了作为 I/O 端口引脚使用外，这些引脚还有输入模拟信号给 A/D 转换器的功能。这些引脚能以 1 位为单位设置成以下的工作模式。

(1) 端口模式

端口 2 是一个 4 位 I/O 端口，可通过端口模式寄存器 2（PM2）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 2（PU2）设置为使用内部上拉电阻。

(2) 控制模式

P20 至 P23 有作为 A/D 转换器的模拟输入引脚（ANI0 至 ANI3）的功能。当作为模拟输入引脚使用时，请参考 **10.6 A/D 转换器的注意事项 (5) ANI0/P20 ～ ANI3/P23**。

2.2.3 P30 ～ P34（端口 3）

端口 3 是一个 4 位 I/O 端口。除了作为 I/O 端口引脚使用外，P30 和 P31 引脚还能作为定时器信号的输入/输出引脚使用，也能输入外部中断请求信号。

P34 是一个输入端口。这个引脚也能作为 $\overline{\text{RESET}}$ 引脚使用。复用功能的详细设置，请参考**第十八章 可选字节**。当 P34 作为输入端口使用时，通过外部电阻上拉 P34 引脚。

P30 至 P33 能以 1 位为单位设置成以下的工作模式。

(1) 端口模式

端口 4 是一个 4 位 I/O 端口，可通过端口模式寄存器 3（PM3）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 3（PU3）设置为使用内部上拉电阻。

P34 是一个输入端口。

(2) 控制模式

P30 和 P31 引脚能作为定时器信号的输入/输出引脚使用，也能输入一个外部中断请求信号。

(a) INTP0 和 INTP2

这是外部中断请求输入引脚，可指定有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) TI000

TI000 将外部事件计数时钟输入到 16 位定时器/计数器 00，或者将捕捉触发信号输入到 16 位定时器/计数器 00 的捕捉寄存器(CR000 和 CR010)。

(c) TI010

TI010 将捕捉触发信号输入到 16 位定时器/计数器 00 的捕捉寄存器(CR000)。

(d) TO00
TO00 从 16 位定时器/计数器 00 输出信号。

2.2.4 P40 ～ P47 (端口 4)

P40 至 P47 作为 8 位 I/O 端口使用。除了作为 I/O 端口引脚使用外，P41 至 P44 也有输出定时器信号，输入外部中断请求信号和输入/输出串行接口的数据的功能。

这些引脚能以 1 位为单位设置成以下的工作模式。

(1) 端口模式

P40 至 P47 是一个 8 位 I/O 端口，可通过端口模式寄存器 4（PM4）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 4（PU4）设置为使用内部上拉电阻。

(2) 控制模式

P41 至 P44 有输出定时器信号，输入外部中断请求信号和输入/输出串行接口的数据的功能。

(a) INTP1 和 INTP3
这是外部中断请求输入引脚，可指定有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) TOH1
这是 8 位定时器 H1 的输出引脚。

(c) TxD6
这个引脚从异步串行接口输出串行数据。

(d) RxD6
这个引脚输入串行数据到异步串行接口。

2.2.5 P120 ～ P123 (端口 12)

P120 至 P123 是一个 4 位 I/O 端口。

这个端口可通过端口模式寄存器 12（PM12）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 12（PU12）设置 P120 和 P123 为使用内部上拉电阻。

P121 和 P122 也能复用为 X1 和 X2 引脚。对复用功能的详细描述，请参见**第十八章　可选字节**。

注意事项　　P121/X1 和 P122/X2 引脚复位期间为下拉状态。

2.2.6 P130 (端口 13)

端口 13 为 1 位输出端口。

2.2.7 $\overline{\text{RESET}}$

这个引脚输入低有效的系统复位信号。

2.2.8 X1 和 X2

X1 输入时钟的振荡器连接。

X1 和 X2 也能复用为 P121 和 P122 引脚，对复用功能的详细描述，请参见**第十八章 可选字节**。

注意事项 P121/X1 和 P122/X2 引脚复位后为下拉状态。

2.2.9 AVREF

这个引脚输入 A/D 转换器的参考电压，当不使用 A/D 转换器时，直接与 VDD 连接。

2.2.10 AVSS

这是 A/D 转换器的地电平引脚，无论是否使用 A/D 转换器，该引脚须与 VSS 引脚保持相同电位。

2.2.11 VDD

这是电源电压引脚。

2.2.12 VSS

这是地引脚。

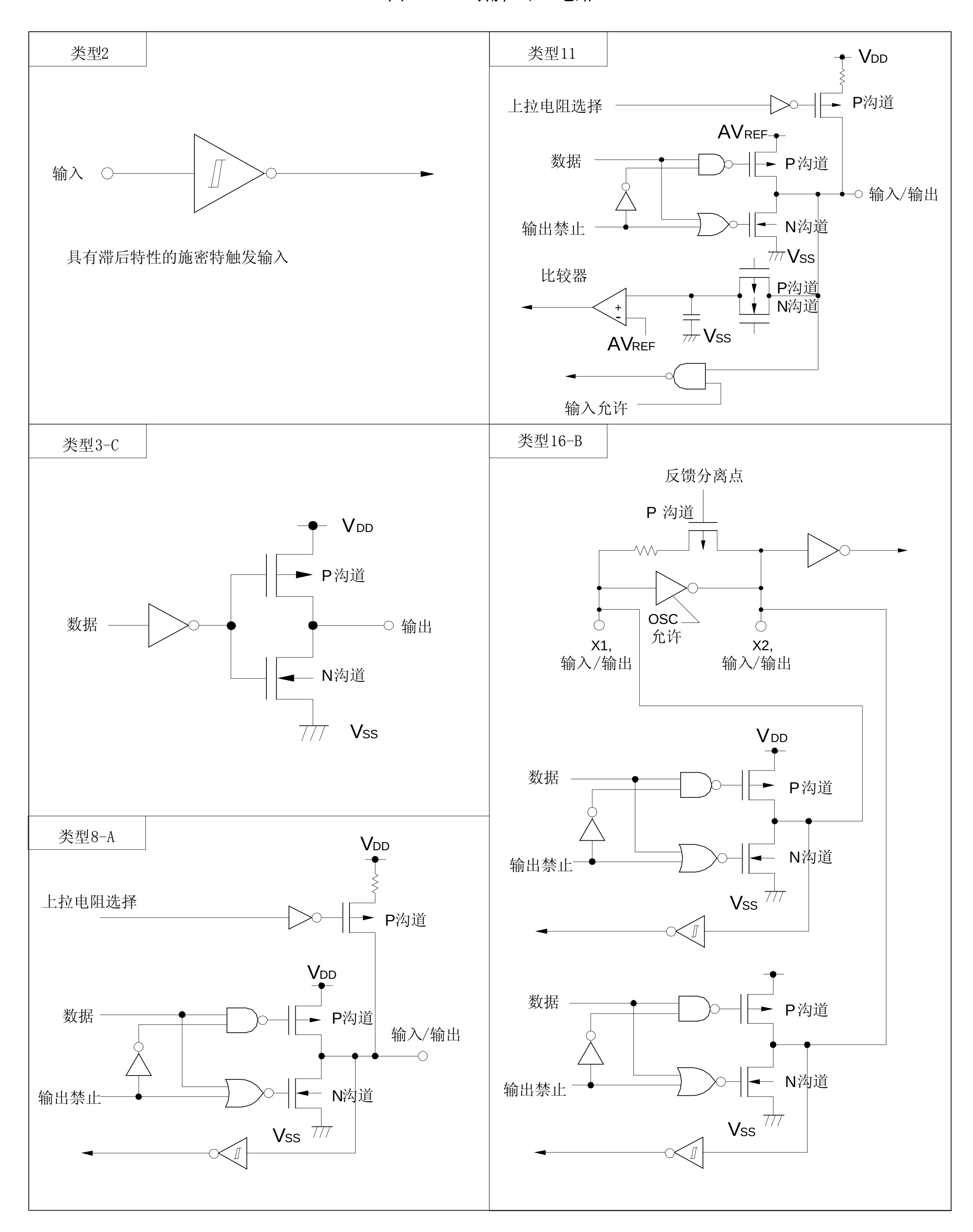
2.3 引脚 I/O 电路和未使用引脚的连接方式

表 2-1 给出了每个引脚的 I/O 电路类型和未使用引脚的连接方式。
每种类型的 I/O 电路的结构，请参照 图 2-1。

表 2-1. 引脚 I/O 电路的类型和未使用引脚的连接方式

引脚名称	I/O 电路类型	I/O	未使用引脚的连接方式
P00 ～ P03	8-A	I/O	输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS} 输出: 悬空
P20/ANI0 ～ P23/ANI3	11		输入: 串接电阻后单独连接到 AV_{REF} 或者 V_{SS} 输出: 悬空
P30/TI000/INTP0	8-A		输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS} 输出: 悬空
P31/TI010/TO00/INTP2			
P32 和 P33			
P34/$\overline{RESET}$	2	输入	串接电阻后连接到 V_{DD}
P40	8-A	I/O	输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS} 输出: 悬空
P41/INTP3			
P42/TOH1			
P43/TxD6/INTP1			
P44/RxD6			
P45 ～ P47			
P120			
P121/X1	16-B		输入: 串接电阻后单独连接到 V_{SS} 输出: 悬空
P122/X2			
P123	8-A		输入: 串接电阻后单独连接到 V_{DD} 或者 V_{SS} 输出: 悬空
P130	3-C	输出	悬空
AV_{REF}	–	–	直接连接到 V_{DD}.
AV_{SS}	–	–	直接连接到 V_{SS}.

图 2-1. 引脚 I/O 电路


类型2
输入
具有滞后特性的施密特触发输入
类型11
VDD
上拉电阻选择
P沟道
AVREF
数据
P沟道
输入/输出
输出禁止
N沟道
VSS
比较器
P沟道
N沟道
AVREF
VSS
输入允许
类型3-C
VDD
P沟道
数据
输出
N沟道
VSS
类型16-B
反馈分离点
P 沟道
OSC
允许
X1,
输入/输出
X2,
输入/输出
VDD
数据
P沟道
输出禁止
N沟道
VSS
数据
P沟道
输出禁止
N沟道
VSS
类型8-A
VDD
上拉电阻选择
P沟道
VDD
数据
P沟道
输入/输出
输出禁止
N沟道
VSS

3.1 存储空间

78K0S/KB1+可以访问大小为 64KB 的存储空间。图 3-1 至 3-2 显示了存储空间映射图。

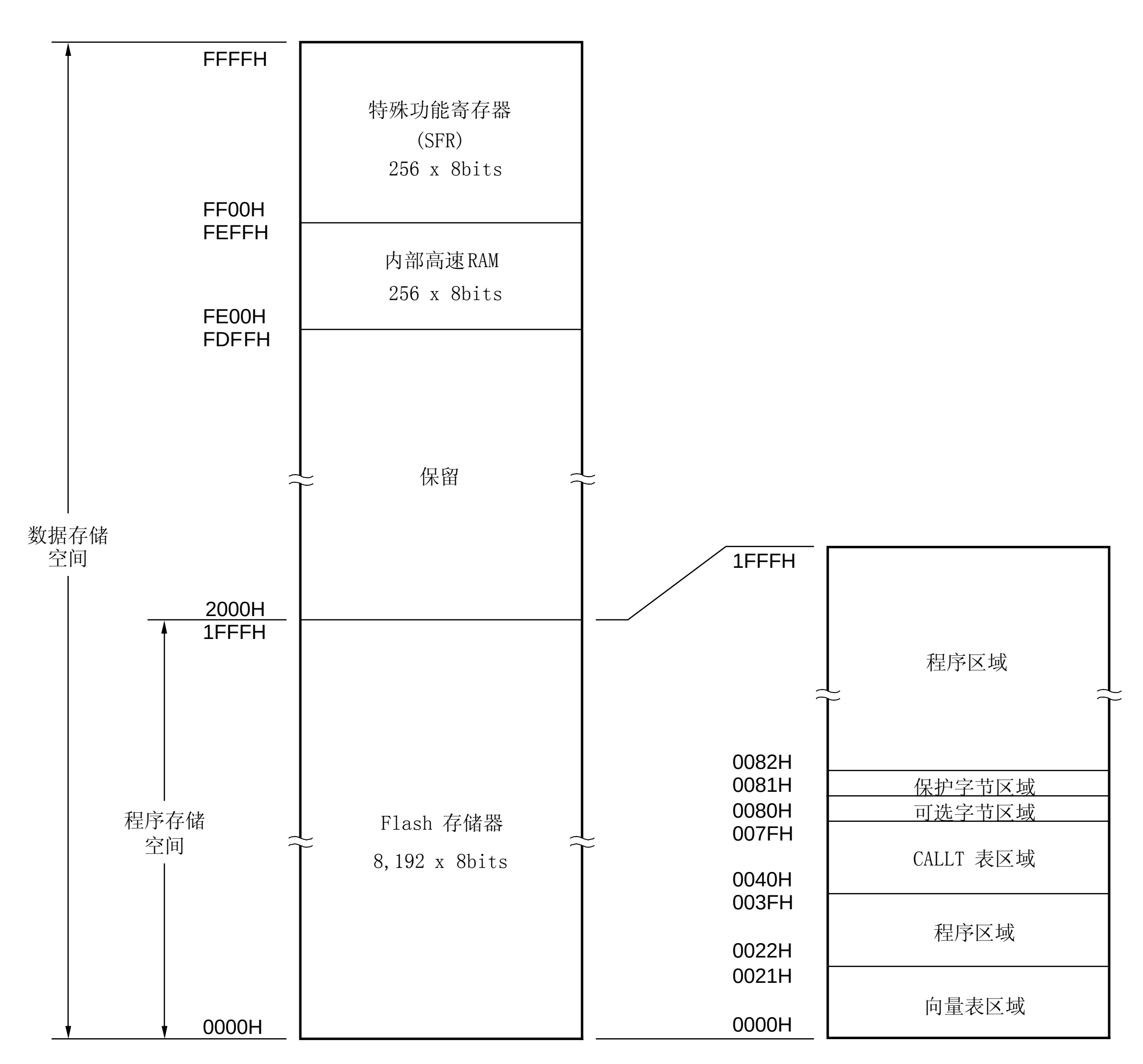
图 3-1. 存储空间映射图（μPD78F9232）

FFFFH
特殊功能寄存器
(SFR)
256 x 8bits
FF00H
FEFFH
内部高速RAM
256 x 8bits
FE00H
FDFFH
保留
数据存储空间
1000H
0FFFH
程序存储空间
Flash 存储器
4,096 x 8bits
0000H

0FFFH
程序区域
0082H
0081H
保护字节区域
0080H
可选字节区域
007FH
CALLT 表区域
0040H
003FH
程序区域
0022H
0021H
向量表区域
0000H

备注 可选字节和保护字节各为 1 字节。

图 3-2. 存储空间映射图 (μPD78F9234)



备注 可选字节和保护字节各为 1 字节。

3.1.1 内部程序存储空间

内部程序存储空间用于存储程序和表格数据。一般通过程序计数器(PC)来寻址。
78K0S/KB1+产品提供了以下容量的内部 ROM（或 FLASH 存储器）。

表 3-1. 内部 ROM 容量

产品编号	内部 ROM	
	结构	容量
μPD78F9232	Flash 存储器	4,096 × 8 bits
μPD78F9234		8,192 × 8 bits

内部程序存储空间主要分为以下几个区域。

(1) 向量表区域

从 0000H 到 0021H 总共 34 字节作为向量表区域。在向量表中存放的是根据复位信号输入或产生的中断请求进行转移的程序起始地址。
对于一个 16 位地址，低 8 位存放在偶地址，高 8 位存放在奇地址。

表 3-2. 向量表

向量表地址	中断请求	向量表地址	中断请求
0000H	复位输入	0012H	INTAD
0006H	INTLVI	0016H	INTP2
0008H	INTP0	0018H	INTP3
000AH	INP1	001AH	INTTM80
000CH	INTTMH1	001CH	INTSRE6
000EH	INTTM000	001EH	INTSR6
0010H	INTTM010	0020H	INTST6

注意事项 向量表地址 0014H 不对应任何中断源。

(2) CALLT 指令表区域

0040H 至 007FH 共 64 字节区域，可存放 1 字节调用指令（CALLT）的子程序入口地址。

(3) 可选字节区域

可选字节区域是地址 0080H 的 1 字节区域。若要了解细节，参考 **第十八章 可选字节。**

(4) 保护字节区域

保护字节区域是地址 0081H 的 1 字节区域。 若要了解细节，参考 **第十九章 FLASH 存储器。**

3.1.2 内部数据存储空间

μPD78F9232 和μPD78F9234 提供 256 字节的内部高速 RAM。
内部高速 RAM 也可以被用作堆栈存储器。

3.1.3 特殊功能寄存器（SFR）区域

片内外围硬件的特殊功能寄存器（SFRs） 被分配在FF00H ～ FFFFH区域内（参见 表 3-3）。

3.1.4 数据存储器寻址

78K0S/KB1+提供宽范围的寻址方式使存储器操作更有效率。包括数据存储器和特殊功能寄存器（SFR）的存储区间（FE00H ～ FEFFH ）能够通过与其功能相一致的唯一寻址模式进行访问。 图 3-3 至 3-4 举例说明数据存储空间寻址。

图 3-3. 数据存储器寻址 （μPD78F9232）

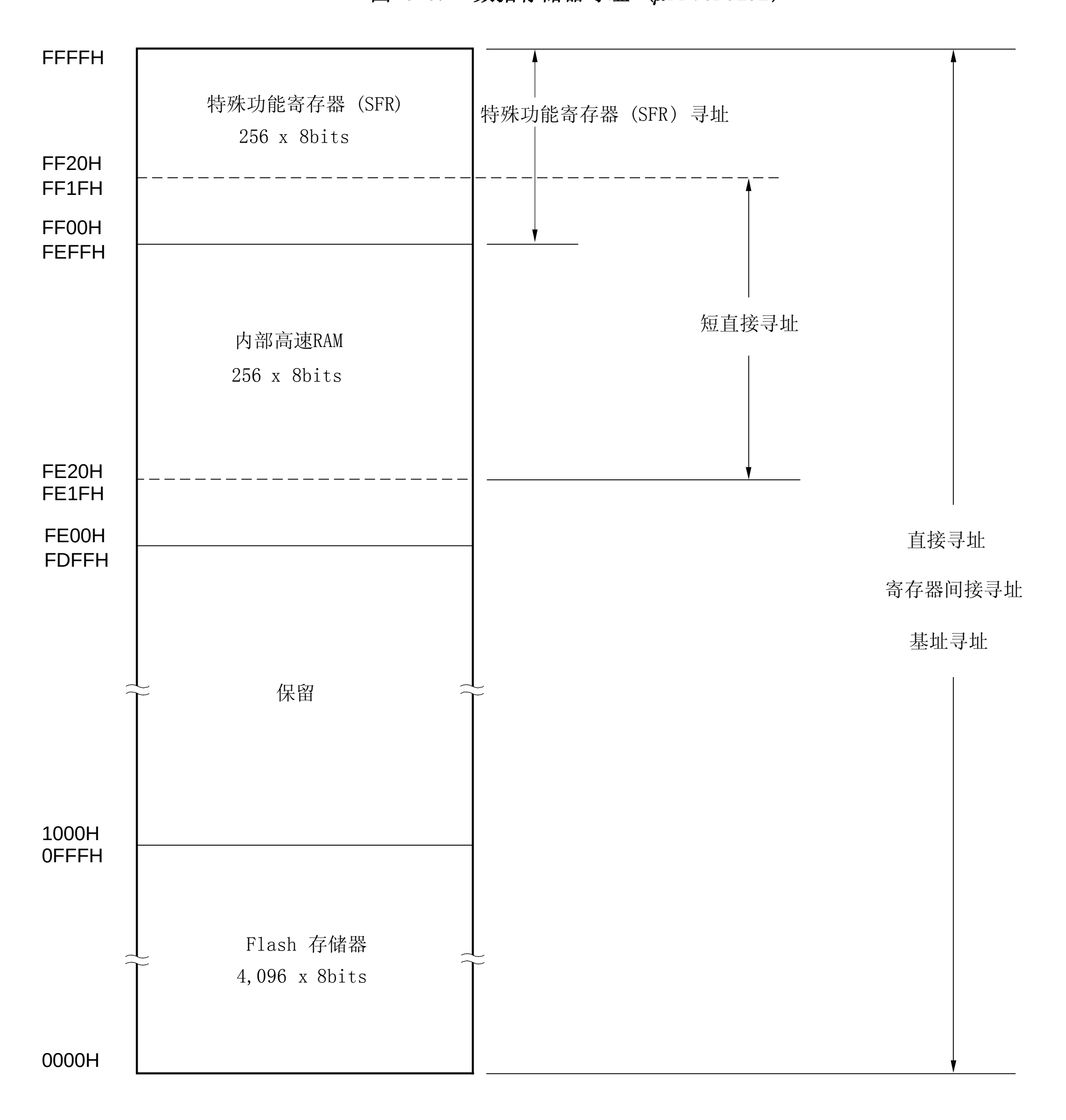
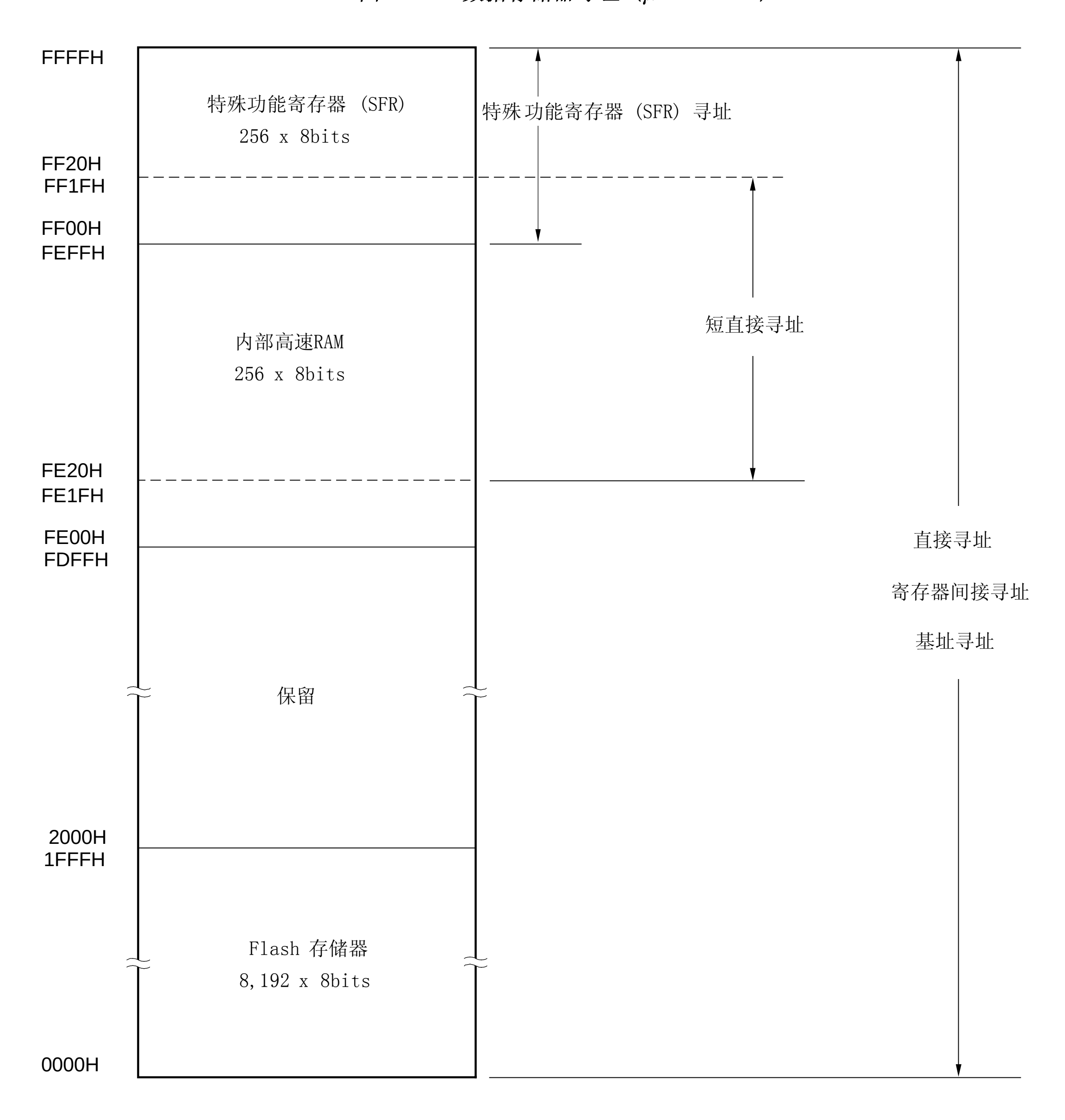


图 3-4. 数据存储器寻址（μPD78F9234）



3.2 处理器寄存器

78K0S/KB1+包含以下几种片内处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序执行顺序的状态和堆栈空间。控制寄存器包括程序计数器（PC）、程序状态字（PSW）和堆栈指针（SP）。

(1) 程序计数器 (PC)

程序计数器是一个 16 位寄存器，用于存放下一条即将执行的程序的地址信息。

在正常操作中，根据获取的指令字节数，程序计数器（PC）的值会自动累加。当执行分支指令时，则设置为立即数或寄存器的值。

$\overline{\text{RESET}}$ 信号产生时，把复位向量表地址为 0000H 和 0001H 的值赋给程序计数器。

图 3-5. 程序计数器结构

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

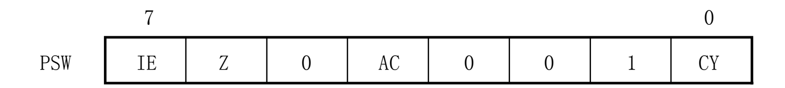
(2) 程序状态字 (PSW)

程序状态字（PSW）是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行设置或复位。

根据中断请求的产生或 PUSH PSW 指令执行，程序状态字内容自动入栈；通过执行 RETI 和 POP PSW 指令，程序状态字的值自动出栈。

$\overline{\text{RESET}}$ 信号输入将程序状态字的值设定为 02H 。

图 3-6. 程序状态字结构



(a) 中断允许标志 (IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE=0 时，表示不允许中断 (DI)，即禁止了除不可屏蔽中断外的所有中断。

当 IE=1 时，表示允许中断 (EI) ，通过各种中断源的中断屏蔽标志来控制响应中断请求。

当执行 DI 指令或中断请求得到响应时，该标志复位（0）；当执行 EI 指令时，该标志置为 1。

(b) 0 标志 (Z)

当操作结果为 0 时，该标志置 1，其它情况下置 0。

(c) 辅助进位标志 (AC)

如果操作结果在第 3 位产生进位或借位，该标志置 1，其它情况下置 0。

(d) 进位标志 (CY)

该标志存储的是在执行加减指令时产生的上溢或下溢。它也存储移位指令执行中的移出值，还可以在位操作指令执行中作为位累加器使用。

(3) 堆栈指针 (SP)

这是一个 16 位寄存器，用来存放内部堆栈区的起始地址。只有内部高速 RAM 区才能被设置为堆栈区。

图 3-7. 堆栈指针结构

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

在向堆栈写（存）内容时，堆栈指针累减，而从堆栈中读出（恢复）内容时，堆栈指针累加。

堆栈的数据存储/恢复操作过程如图 3-8 和 3-9 所示。

注意事项 由于复位输入使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。

图 3-8. 将数据存入堆栈

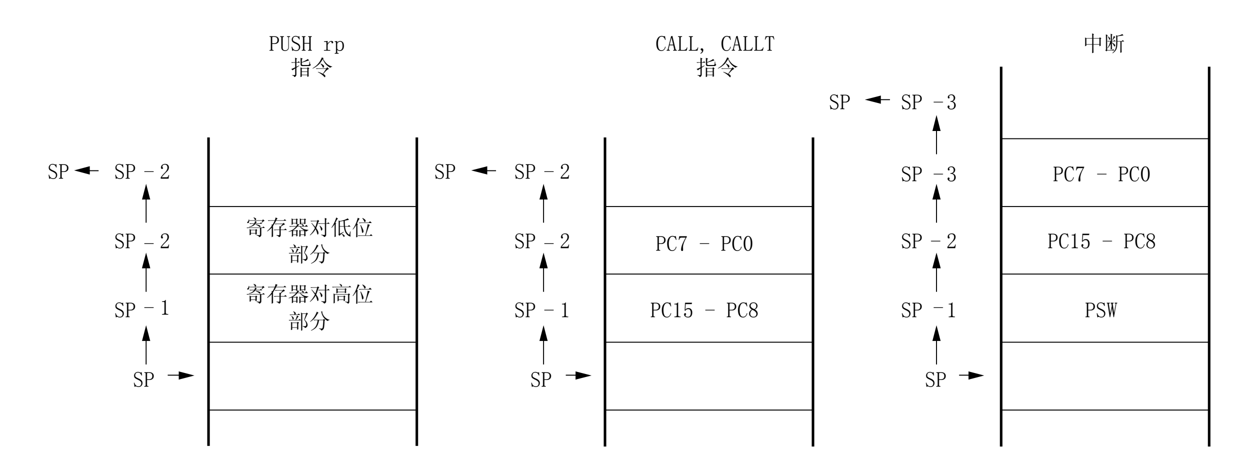
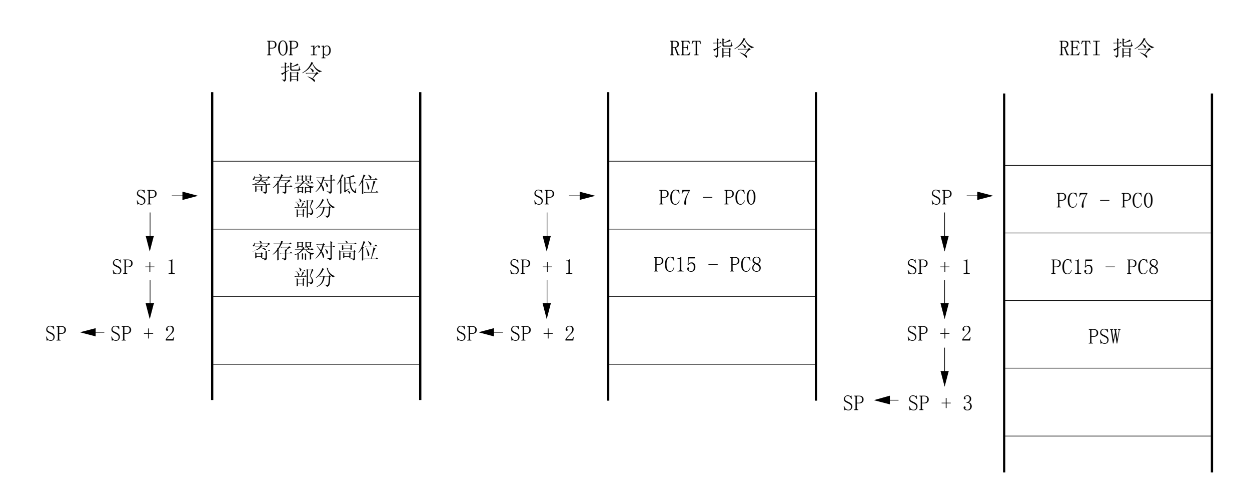


图 3-9. 从堆栈读出数据



3.2.2 通用寄存器

通用寄存器包含了 8 个 8 位寄存器(X, A, C, B, E, D, L 和 H)。

此外，每个寄存器都可以当作一个 8 位寄存器使用，两个成对的 8 位寄存器可以当作一个 16 位寄存器使用(AX, BC, DE 和 HL)。

描述通用寄存器可以使用功能名称 (X, A, C, B, E, D, L, H, AX, BC, DE 和 HL) 和绝对名称 (R0 ～ R7 和 RP0 ～ RP3)。

图 3-10. 通用寄存器结构

(a) 绝对名称

16位处理 (15–0)	8位处理 (7–0)
RP3	R7
	R6
RP2	R5
	R4
RP1	R3
	R2
RP0	R1
	R0

(b) 功能名称

16位处理 (15–0)	8位处理 (7–0)
HL	H
	L
DE	D
	E
BC	B
	C
AX	A
	X

3.2.3 特殊功能寄存器 (SFRs)

与通用寄存器不同，每一个特殊功能寄存器都有其特定的功能。

从 FF00H 到 FFFFH 总共 256 字节的区域分配给特殊功能寄存器。

特殊功能寄存器可以像通用寄存器那样用操作指令、转移指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的单元也不同，可以是 1 位、8 位和 16 位。

每个操作单元描述如下。

- 1 位操作
  1 位操作指令的操作数(sfr.bit)被描述为汇编程序的保留符号。
  该操作也可以由一个地址来定义。

- 8 位操作
  8 位操作指令的操作数(sfr)被描述为汇编程序的保留符号。
  该操作也可以由一个地址来定义。

- 16 位操作
  16 位操作指令的操作数(sfrp)被描述为汇编程序的保留符号。
  该操作也可以由一个地址(偶地址)来定义。

表 3-3 为特殊功能寄存器列表。表中术语的含义如下。

- 符号
  符号表示特殊功能寄存器的地址，在 RA78K0S 中是保留字，并在 CC78K0S 中通过使用#pragma sfr 伪指令来声明它们作为 sfr 变量使用。如果使用汇编器或集成调试器时，这些符号可以被用作指令操作数。

- R/W
  表示特殊功能寄存器可读或可写。
  R/W: 可读写
  R:　只读
  W:　只写

- 可同时位操作的数量
  特殊功能寄存器可以位操作的单元(1, 8, 和 16)。

- 复位后
  表示复位信号输入后特殊功能寄存器的状态。

表 3-3. 特殊功能寄存器 (1/3)

地址	特殊功能寄存器(SFR)名称	符号		R/W	可同时操作的位单元数目			复位后
					1 位	8 位	16 位	
FF00H	端口寄存器 0	P0		R/W 注 1	√	√	–	00H
FF02H	端口寄存器 2	P2			√	√	–	
FF03H	端口寄存器 3	P3			√	√	–	
FF04H	端口寄存器 4	P4			√	√	–	
FF0CH	端口寄存器 12	P12			√	√	–	
FF0DH	端口寄存器 13	P13			√	√	–	
FF0EH	8 位定时器 H 比较寄存器 01	CMP01		R/W	–	√	–	
FF0FH	8 位定时器 H 比较寄存器 11	CMP11			–	√	–	
FF10H	16 位乘法结果存储寄存器 L	MUL0L	MUL0	R	–	√	√	不确定
FF11H	16 位乘法结果存储寄存器 H	MUL0H			–	√		
FF12H	16 位定时器计数器 00	TM00			–	–	√注 2	0000H
FF13H								
FF14H	16 位定时器捕捉/比较寄存器 000	CR000		R/W	–	–	√注 2	0000H
FF15H								
FF16H	16 位定时器捕捉/比较寄存器 010	CR010			–	–	√注 2	0000H
FF17H								
FF18H	10 位 A/D 转换结果寄存器	ADCR		R	–	–	√注 2	不确定
FF19H								
FF1AH	8 位 A/D 转换结果寄存器	ADCRH			–	√	–	
FF20H	端口模式寄存器 0	PM0		R/W	√	√	–	FFH
FF22H	端口模式寄存器 2	PM2			√	√	–	
FF23H	端口模式寄存器 3	PM3			√	√	–	
FF24H	端口模式寄存器 4	PM4			√	√	–	
FF2CH	端口模式寄存器 12	PM12			√	√	–	
FF30H	上拉电阻选择寄存器 0	PU0			√	√	–	00H
FF32H	上拉电阻选择寄存器 2	PU2			√	√	–	
FF33H	上拉电阻选择寄存器 3	PU3			√	√	–	
FF34H	上拉电阻选择寄存器 4	PU4			√	√	–	
FF3CH	上拉电阻选择寄存器 12	PU12			√	√	–	
FF48H	看门狗定时器模式寄存器	WDTM			–	√	–	67H
FF49H	看门狗定时器使能寄存器	WDTE			–	√	–	9AH
FF50H	低电压检测寄存器	LVIM			√	√	–	00H 注 3
FF51H	低电压检测电平选择寄存器	LVIS			–	√	–	
FF54H	复位控制标志寄存器	RESF		R	–	√	–	00H 注 4
FF58H	低速 Ring-OSC 模式寄存器	LSRCM		R/W	√	√	–	00H

注 1. P34 只能为输入端口。
2. 仅能通过短直接寻址方式进行 16 位访问。
3. 只有在 LVI 引起的复位后才能保持。
4. 随复位原因的不同而不同。

表 3-3. 特殊功能寄存器 (2/3)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可同时操作的位单元数目			复位后
				1 位	8 位	16 位	
FF60H	16 位定时器模式控制寄存器 00	TMC00	R/W	√	√	–	00H
FF61H	预分频模式寄存器 00	PRM00		√	√	–	
FF62H	捕捉/比较控制寄存器 00	CRC00		√	√	–	
FF63H	16 位定时器输出控制寄存器 00	TOC00		√	√	–	
FF70H	8 位定时器 H 模式寄存器 1	TMHMD1		√	√	–	
FF80H	A/D 转换器模式寄存器	ADM		√	√	–	
FF81H	模拟输入通道指定寄存器	ADS		√	√	–	
FF84H	端口模式控制寄存器 2	PMC2		√	√	–	
FF8CH	输入开关控制寄存器	ISC		√	√	–	
FF90H	异步串行接口操作模式寄存器 6	ASIM6		√	√	–	01H
FF92H	接收缓冲寄存器 6	RXB6	R	–	√	–	FFH
FF93H	异步串行接口接收错误状态寄存器 6	ASIS6		–	√	–	00H
FF94H	发送缓冲寄存器 6	TXB6	R/W	–	√	–	FFH
FF95H	异步串行接口发送状态寄存器 6	ASIF6	R	–	√	–	00H
FF96H	时钟选择寄存器 6	CKSR6	R/W	–	√	–	
FF97H	波特率发生器控制寄存器 6	BRGC6		–	√	–	FFH
FF98H	异步串行接口控制寄存器 6	ASICL6		√	√	–	16H
FFA0H	Flash 保护命令寄存器	PFCMD	W	–	√	–	不确定
FFA1H	Flash 状态寄存器	PFS	R/W	√	√	–	00H
FFA2H	Flash 编程模式控制寄存器	FLPMC		–	√	–	不确定
FFA3H	Flash 编程命令寄存器	FLCMD		√	√	–	00H
FFA4H	Flash 地址指针 L	FLAPL		√	√	–	不确定
FFA5H	Flash 地址指针 H	FLAPH		√	√	–	
FFA6H	Flash 地址指针 H 比较寄存器	FLAPHC		√	√	–	00H
FFA7H	Flash 地址指针 L 比较寄存器	FLAPLC		√	√	–	
FFA8H	Flash 写缓冲寄存器	FLW		–	√	–	
FFCCH	8 位定时器模式控制寄存器 80	TMC80		√	√	–	
FFCDH	8 位比较寄存器 80	CR80	W	–	√	–	不确定
FFCEH	8 位定时器计数器 80	TM80	R	–	√	–	00H
FFD0H	乘法数据寄存器 A	MRA0	W	–	√	–	不确定
FFD1H	乘法数据寄存器 B	MRB0		–	√	–	
FFD2H	乘法器控制寄存器 0	MULC0	R/W	√	√	–	00H
FFE0H	中断请求标志寄存器 0	IF0		√	√	–	
FFE1H	中断请求标志寄存器 1	IF1		√	√	–	
FFE4H	中断屏蔽标志寄存器 0	MK0		√	√	–	FFH
FFE5H	中断屏蔽标志寄存器 1	MK1		√	√	–	
FFECH	外部中断模式寄存器 0	INTM0		–	√	–	00H

表 3-3. 特殊功能寄存器 (3/3)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可同时操作的位单元数目			复位后
				1 位	8 位	16 位	
FFEDH	外部中断模式寄存器 1	INTM1		–	√	–	00H
FFF3H	预处理器时钟控制寄存器	PPCC		√	√	–	02H
FFF4H	振荡稳定时间选择寄存器	OSTS		–	√	–	不确定[注]
FFFBH	处理器时钟控制寄存器	PCC		√	√	–	02H

注 复位释放后的振荡稳定时间可通过可选字节选择。若要获得细节，参考 **第十八章 可选字节。**

3.3 指令地址寻址

一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令的字节数，程序计数器(PC)的内容自动增加(每字节+1)。在执行分支指令时，将程序计数器(PC)的内容设置为分支的目的地址，并按以下方式确定地址。（想要了解每条指令的详细信息，请参考78K/0S **系列指令用户手册**（U11047E））。

3.3.1 相对寻址

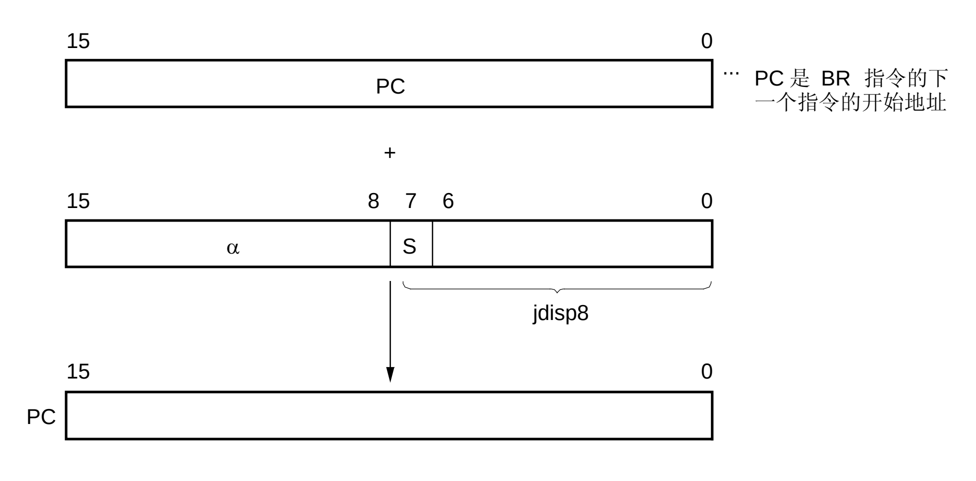
［功能］

将一条指令的 8 位立即数(偏移量：jdisp8)与下一条指令的起始地址相加，结果赋给程序计数器(PC)，然后转向相加结果指向的地址。这个偏移量是带符号位的补码(-128 到+127)，其中第 7 位是符号位。

换句话说，在相对寻址中，分支的范围是从下一条指令起始地址的-128 到+127 之间。

当执行“BR $addr16”指令或条件转移指令时，将实现上述功能。

［图示］



3.3.2 立即寻址

[功能]

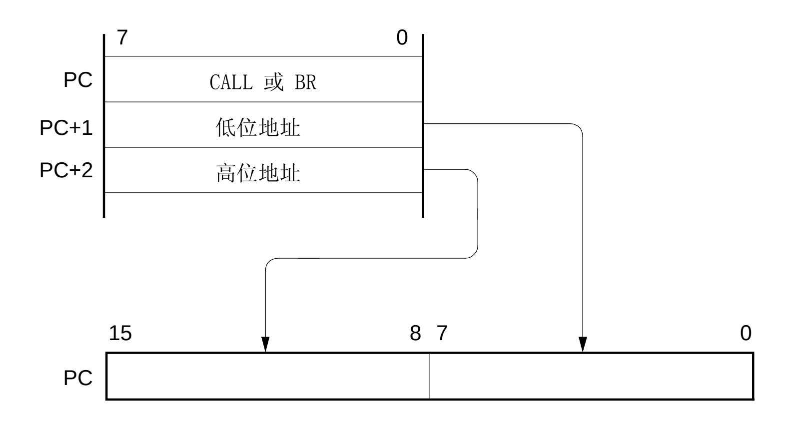
将指令中的立即数赋给程序计数器(PC)，然后转向该地址。

在执行“CALL !addr16”指令和“BR !addr16”指令时，实现此功能。

“CALL !addr16” 和“ BR !addr16”指令的转移地址范围是所有内存空间。

[图示]

CALL !addr16 指令和 BR !addr16 指令



3.3.3 表间接寻址

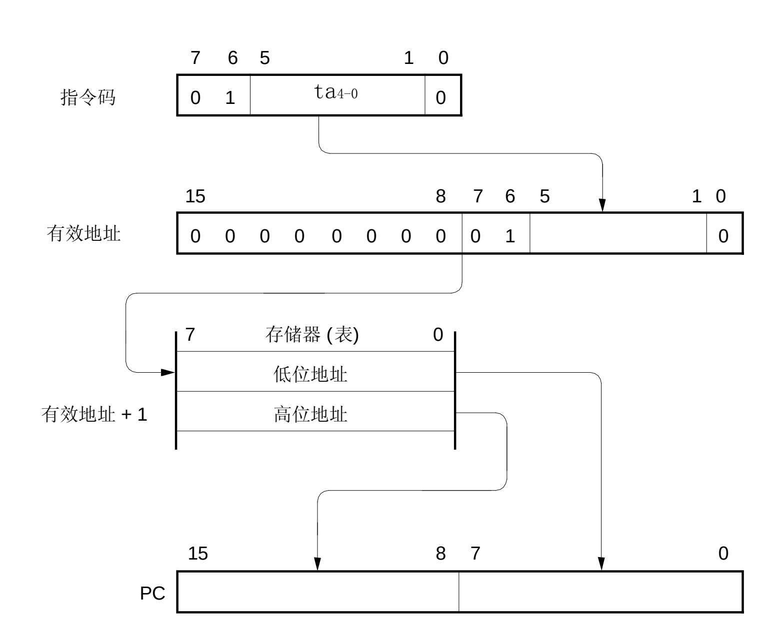
[功能]

通过指令码低 5 位的立即数(从第 1 位到第 5 位)，访问特定存储单元的表的内容(转移目的地址)，并将表的内容赋给程序计数器(PC)，然后转向该地址。

在执行“CALLT [addr5]”指令时，进行表间接寻址。

该指令访问的地址范围是内存表中的 40H 至 7FH 之间所存储的地址，转移地址范围可以是整个存储器空间。

[图示]



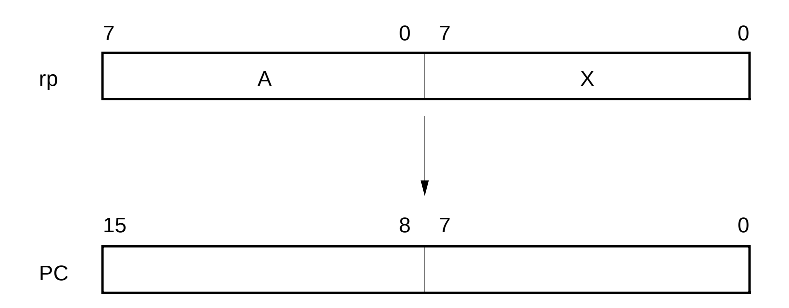
3.3.4 寄存器寻址

［功能］

将一条指令的寄存器对(AX)的内容赋值给程序计数器(PC)，然后转向该地址。

当执行“BR AX”指令时，实现此功能。

［图示］



3.4 操作数地址寻址

以下方法用来规定指令执行期间寄存器寻址和存储器寻址所进行的操作。

3.4.1 直接寻址

[功能]

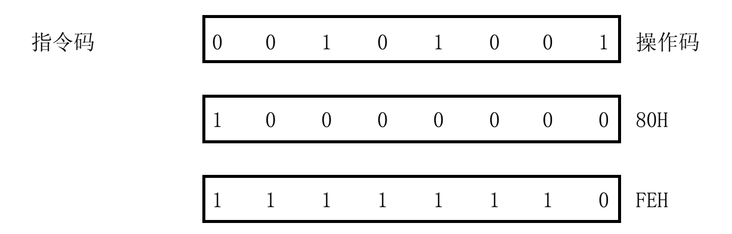
直接寻址方式根据指令中的立即数直接寻址。

[操作数格式]

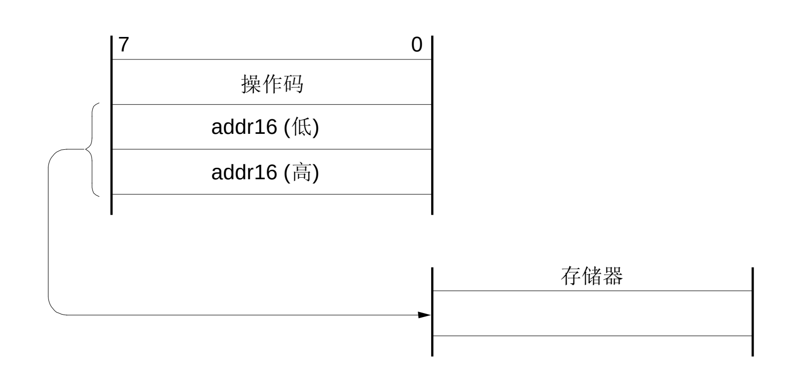
标识符	描述
addr16	标号或 16 位立即数

[举例]

MOV A, !FE80H；将!addr16 设置为 FE80H



[图示]



3.4.2 短直接寻址

[功能]

用指令中的 8 位立即数直接对内存的固定操作区域寻址。

该方式的寻址范围为 FE20H 到 FF1FH 的 256 字节的空间。内部高速 RAM 映射在地址 FE20H 到 FEFFH，特殊功能寄存器 (SFR) 映射在地址 FF00H 到 FF1FH。

采取短直接寻址的特殊功能寄存器(SFR)区域(FF00H 到 FF1FH)是整个特殊功能寄存器 SFR 区域的一部分。

程序中经常访问的端口和定时器计数器的比较寄存器都被映射到该区域。这些特殊功能寄存器(SFR)可以用很少的字节数和时钟数进行操作。

如果 8 位立即数在 20H 到 FFH 之间，则将一个有效地址的第 8 位设置为 0；如果 8 位立即数是在 00H 到 1FH 之间，则把一个有效地址的第 8 位设置为 1。参见下面的[图示] 。

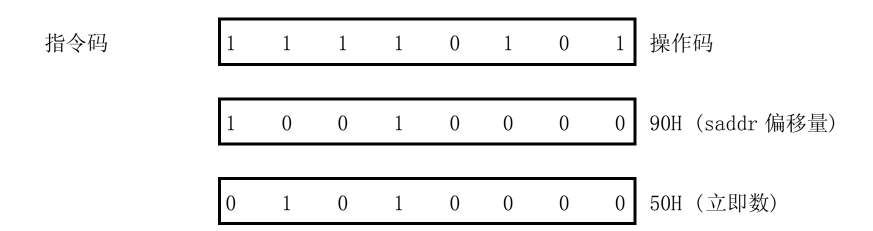
[操作数格式]

标识符	描述
saddr	标号或 FE20H 至 FF1FH 的立即数
saddrp	标号或 FE20H 至 FF1FH 的立即数 (仅使用偶地址)

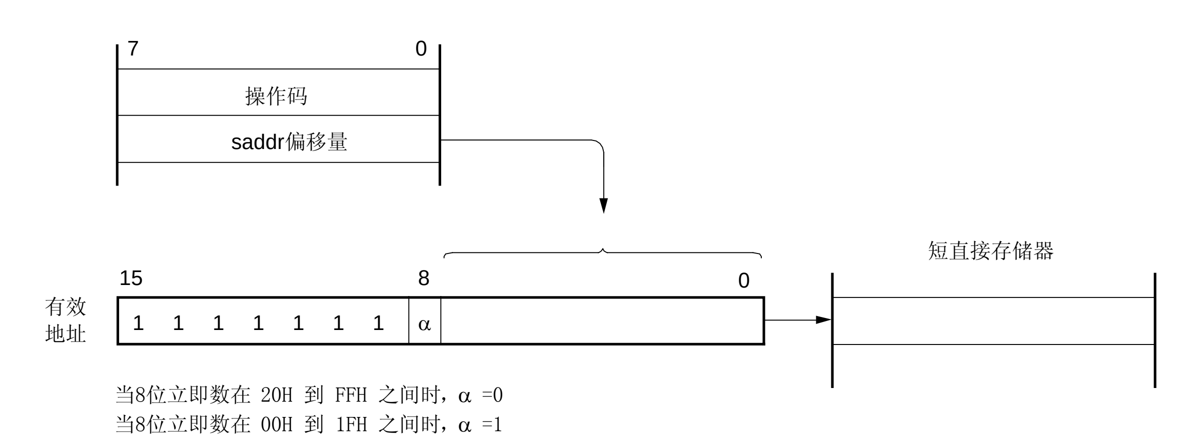
[举例]

EQU DATA1 0FE90H； DATA1 指定为 FE90H 的 saddr 区域

MOV DATA1, #50H；把立即数 50H 赋给 DATA1 单元



[图示]



3.4.3 特殊功能寄存器（SFR）寻址

[功能]

通过指令中的 8 位立即数对内存中的特殊功能寄存器(SFR)区域进行寻址。

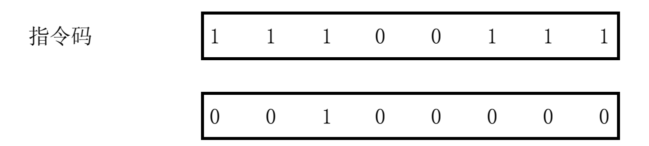
这种寻址方式可以应用到 FF00H 到 FFFFH 共 256 字节的空间。访问 SFR 映射在 FF00H 到 FF1FH 区域的区间需要使用短直接寻址方式。

[操作数格式]

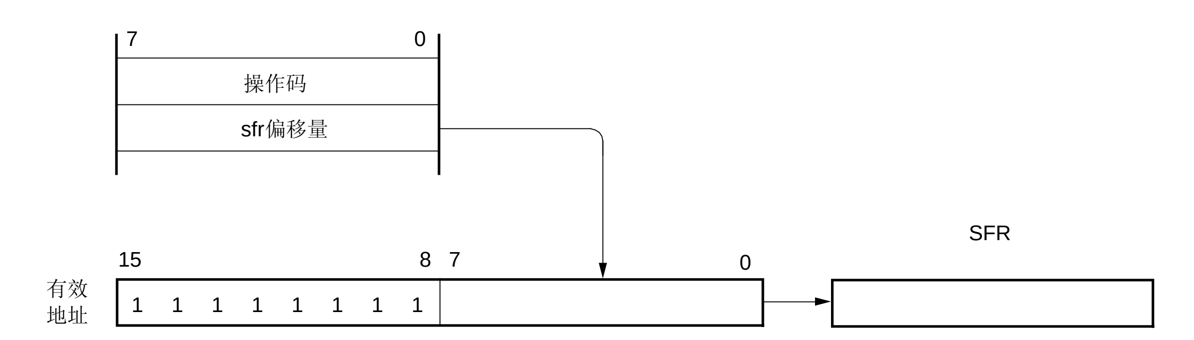
标识符	描述
sfr	特殊功能寄存器名

[举例]

MOV PM0， A； 当选择 PM0 为 sfr 时



[图示]



3.4.4 寄存器寻址

[功能]

通用寄存器可以作为操作数被访问。

由指令中的寄存器标识码和功能名称来指定需要访问的通用寄存器。

当具有下列操作数格式的指令执行时，采用寄存器寻址方式。如果使用 8 位寄存器，则指令码中有 3 位用来表示一个 8 位寄存器。

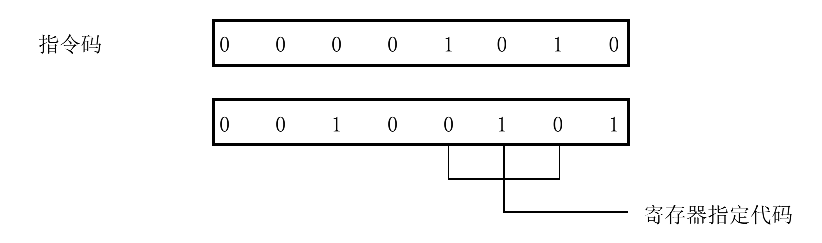
[操作数格式]

标识符	描述
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

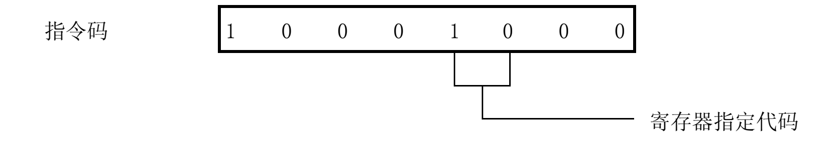
‘r’ 和 ‘rp’ 可用绝对名称 (R0 ～ R7 和 RP0 ～ RP3) 或者功能名称 (X, A, C, B, E, D, L, H, AX, BC, DE, 和 HL)来描述。

[举例]

MOV A， C； 选择寄存器 C 作为 r



INCW DE； 选择寄存器组 DE 作为 rp



3.4.5 寄存器间接寻址

［功能］

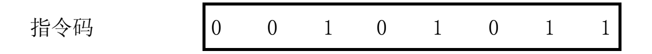
通过把寄存器对中的内容作为操作数来访问存储器。由指令中的寄存器对标识码来指定要访问的寄存器对。可以在所有内存空间使用这种寻址方式。

［操作数格式］

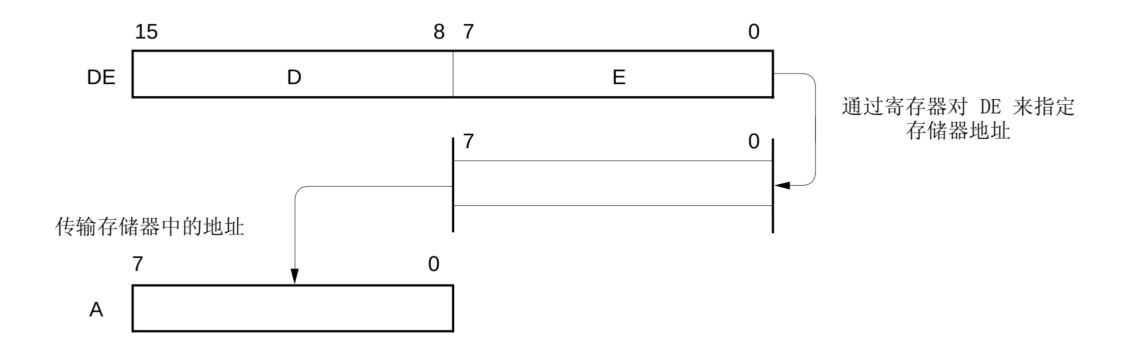
标识符	描述
–	[DE], [HL]

［举例］

MOV A, [DE]; 选择寄存器对 DE 作为操作数



［图示］



3.4.6 基址寻址

[功能]

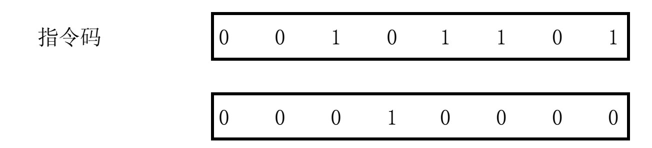
将 8 位立即数加到寄存器对 HL 中，HL 寄存器对作为基地址寄存器，根据相加的结构去寻址。通过将偏移量扩展为 16 位正数，完成加法操作，进位忽略不计。这种寻址方式可以对整个内存空间进行。

[操作数格式]

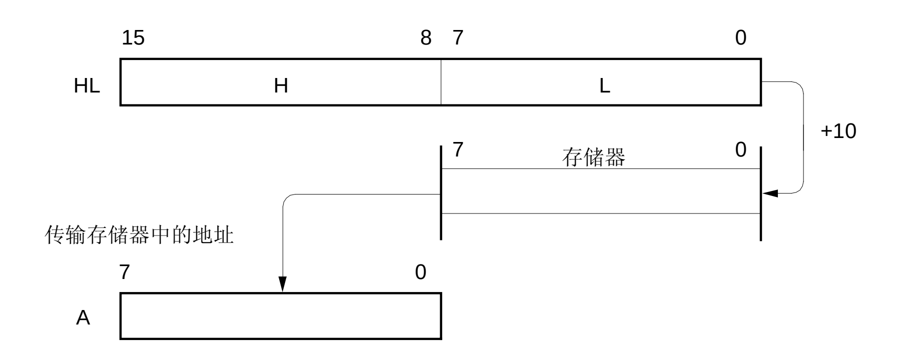
标识符	描述
–	[HL+byte]

[举例]

MOV A, [HL+10H]; byte 的值为 10H



[图示]



3.4.7 堆栈寻址

[功能]

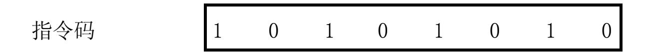
根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。

当执行 PUSH、POP、子程序调用和 RETURN 指令时，或者根据中断请求产生对寄存器进行设置或复位时，将自动使用这种寻址方式。

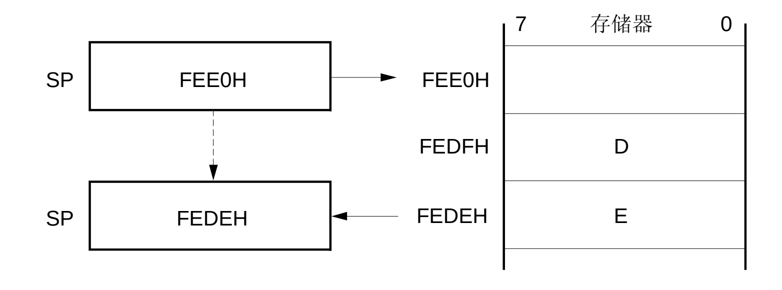
该方式仅能对内部高速 RAM 区域进行寻址。

[举例]

以 PUSH DE 指令为例



[图示]



第四章　端口功能

4.1　端口功能

78K0S/KB1+产品中提供的端口如图 4-1 所示，可被用于不同的控制操作。每个端口功能如表 4-1 所示。
除了作为数字 I/O 端口外，这些端口还有复用功能。需要了解这些端口详细的复用功能，请参阅**第二章 引脚功能**。

图 4-1.　端口功能

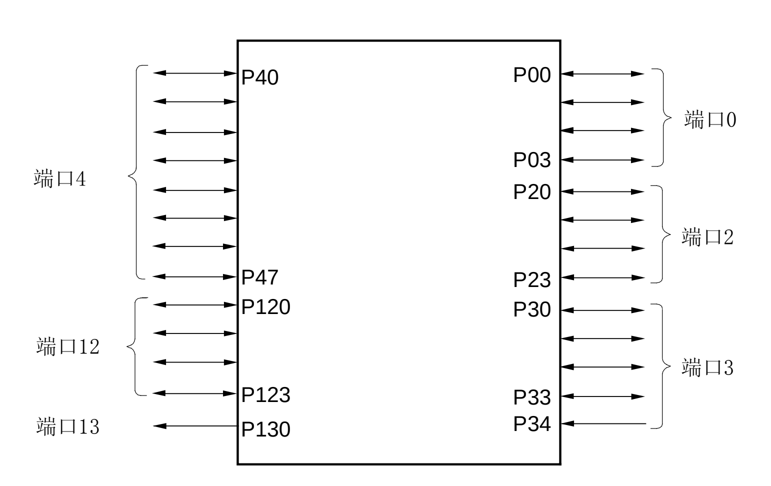


表 4-1.　端口功能

引脚名称	I/O	功能	复位后	复用功能引脚
P00 ～ P03	I/O	端口 0 4 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	–
P20 ～ P23	I/O	端口 2 4 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	ANI0 ～ ANI3
P30	I/O	端口 3 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	TI000/INTP0
P31				TI010/TO00/INTP2
P32				–
P33				–
P34 [注]	输入	仅输入	输入	$\overline{RESET}$ [注]
P40	I/O	端口 4 8 位 I/O 端口 可以位选输入/输出方式 可以通过软件设置连接内部上拉电阻	输入	–
P41				INTP3
P42				TOH1
P43				TxD6/INTP1
P44				RxD6
P45				–
P46				–
P47				–
P120	I/O	端口 12 4 位 I/O 端口 可以位选输入/输出方式 仅 P120 和 P123 能通过软件设置连接内部上拉电阻	输入	–
P121 [注]				X1 [注]
P122 [注]				X2 [注]
P123				–
P130	输出	端口 13 1 位输出端口	输出	–

注　复用功能的详细设置，请参见第十八章 可选字节。

注意事项　P121/X1 和 P122/X2 引脚复位期间为下拉状态。

备注　1. P121 和 P122 引脚在选择高速 Ring-OSC 作为系统时钟时可作它用。
2. P122 在选择外部时钟输入作为系统时钟时可作它用。

4.2　端口配置

端口包括下列硬件单元。

表 4-2.　端口配置

项目	配置
控制寄存器	端口模式寄存器（PM0，PM2，PM3，PM4，PM12） 端口寄存器（P0，P2，P3，P4，P12，P13） 端口模式控制寄存器 2（PMC2） 上拉电阻选择寄存器（PU0，PU2，PU3，PU4，PU12）
端口	总计：26 （CMOS I/O：24，CMOS 输入：1，CMOS 输出：1）
上拉电阻	总计：22

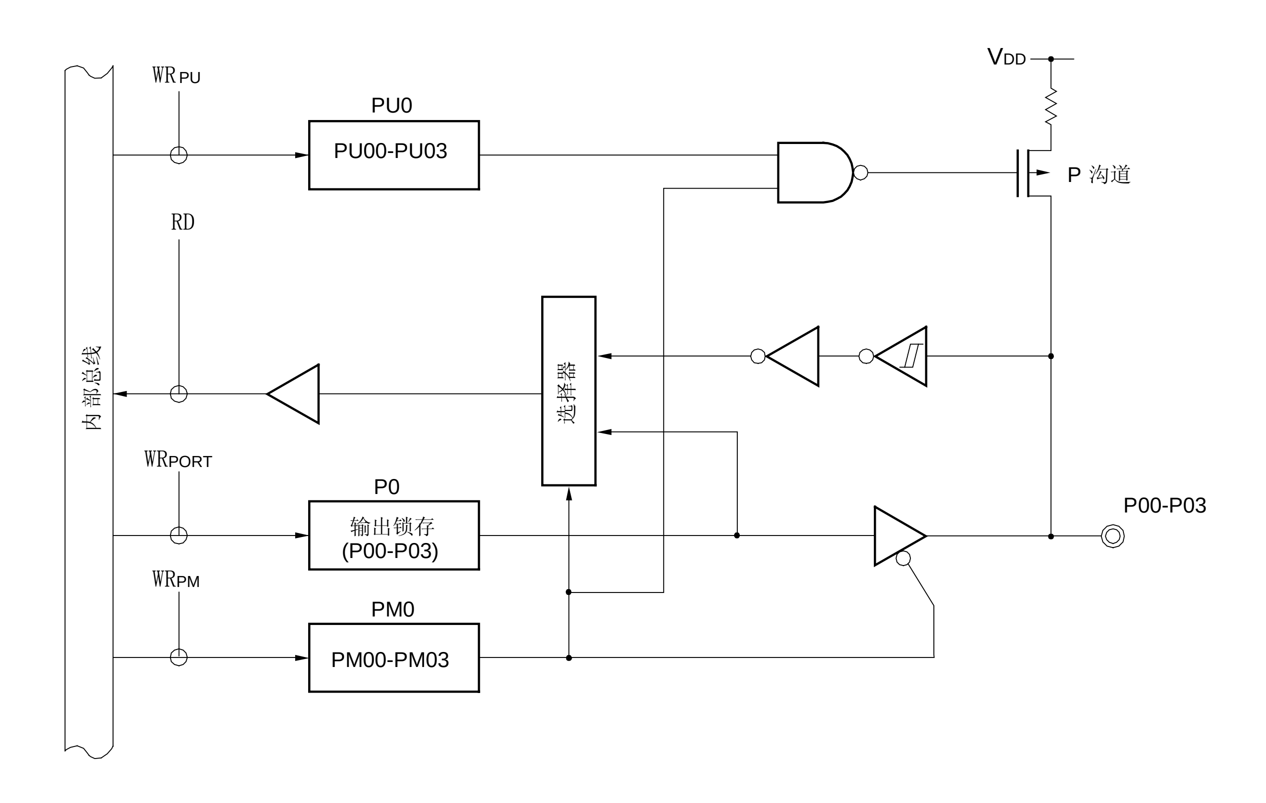
4.2.1　端口 0

端口 0 是一个带输出锁存的 4 位 I/O 端口，可通过端口模式寄存器 0（PM0）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 0（PU0）设置为使用内部上拉电阻。

复位后端口 0 为输入模式。

图 4-2 显示了端口 0 的框图。

图 4-2.　P00 至 P03 的框图



PU0：　上拉电阻选择寄存器 0
P0：　端口寄存器 0
PM0：　端口模式寄存器 0
RD：　读信号
WR××：　写信号

4.2.2 端口 2

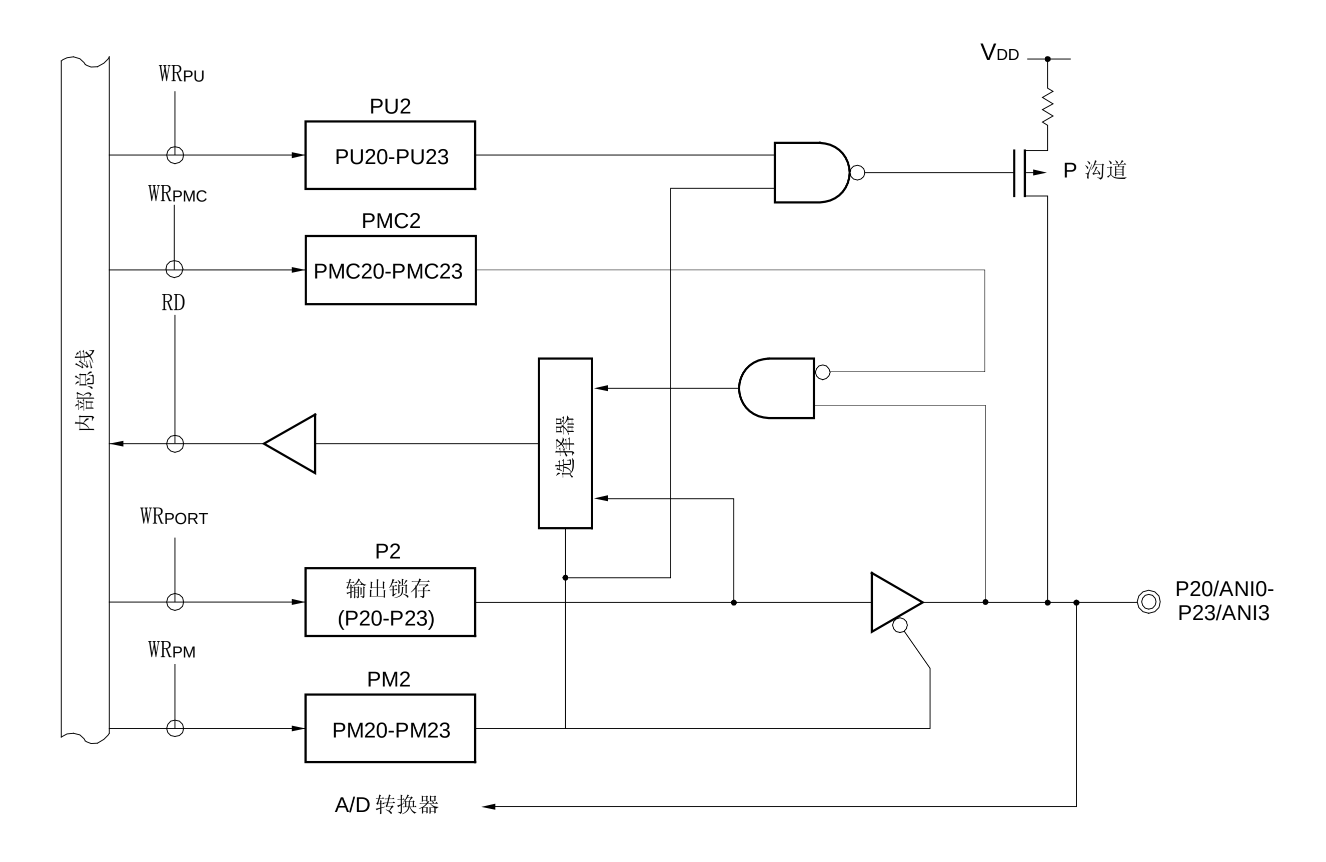
端口 2 是一个带输出锁存的 4 位 I/O 端口。可通过端口模式寄存器 2（PM2）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 2（PU2）设置为使用内部上拉电阻。

这个引脚还能作为 A/D 转换器的模拟输入引脚使用。

复位后端口 2 为输入模式。

图 4-2 显示了端口 2 的框图。

图 4-3. P20 至 P23 的框图



PU2： 上拉电阻选择寄存器 2
P2： 端口寄存器 2
PM2： 端口模式寄存器 2
PMC2： 端口模式控制寄存器 2
RD： 读信号
WR××： 写信号

4.2.3 端口 3

端口 P3 是一个带输出锁存的 4 位 I/O 端口。可通过端口模式寄存器 3（PM3）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 3（PU3）设置为使用内部上拉电阻。

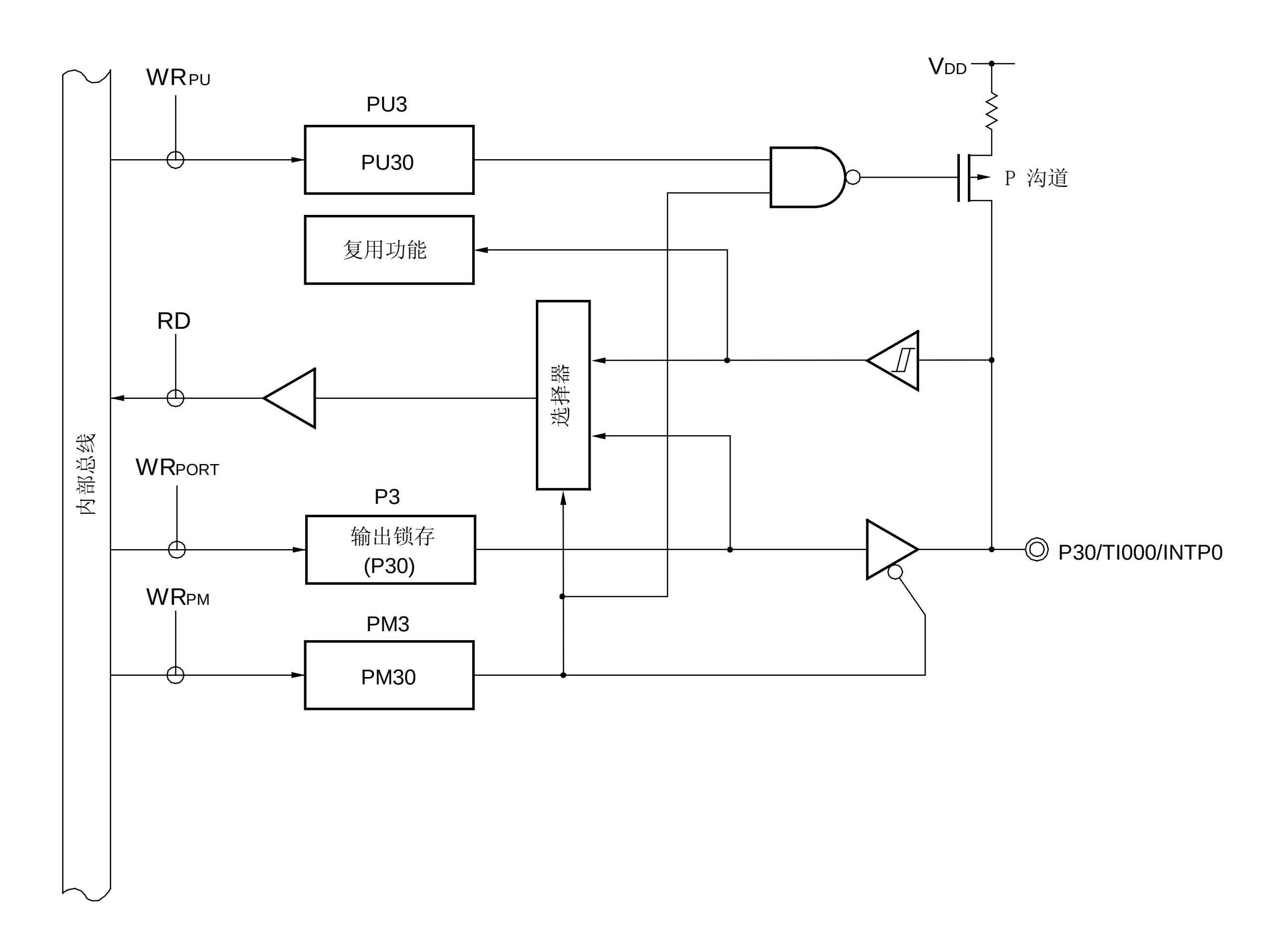
P30 和 P31 引脚还能作为定时器信号的输入/输出引脚使用，也能作为输入外部中断请求信号引脚使用。

复位后端口 3 为输入模式。

P34 引脚只能为 1 位输入端口并能复用为 $\overline{\text{RESET}}$ 引脚。如需了解详细的功能设置，请参见 **第十八章 可选字节**。当 P34 作为输入引脚使用时，通过外部电阻上拉 P34 引脚。

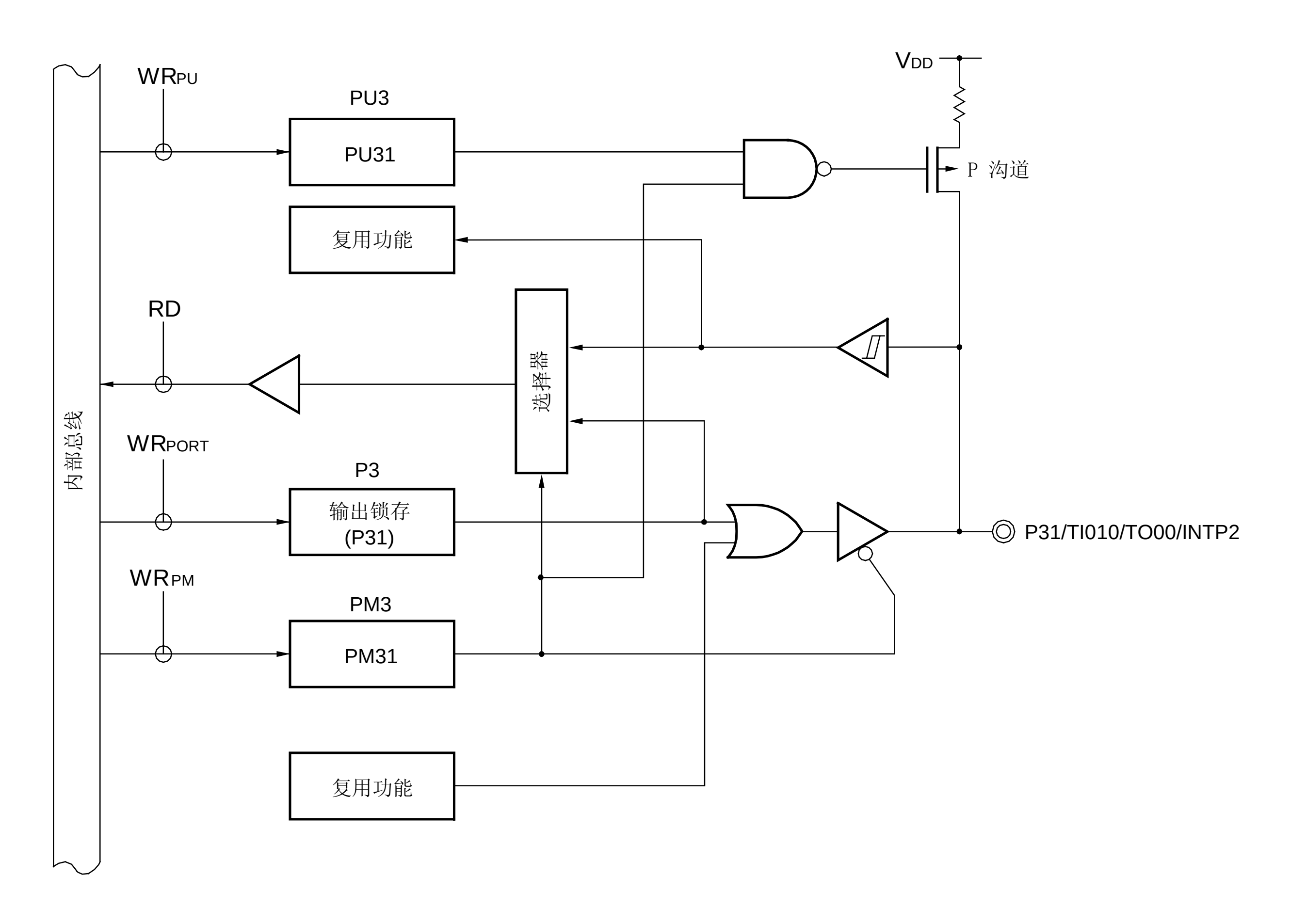
图 4-4 至 4-7 显示了端口 3 的框图。

图 4-4.　P30 的框图



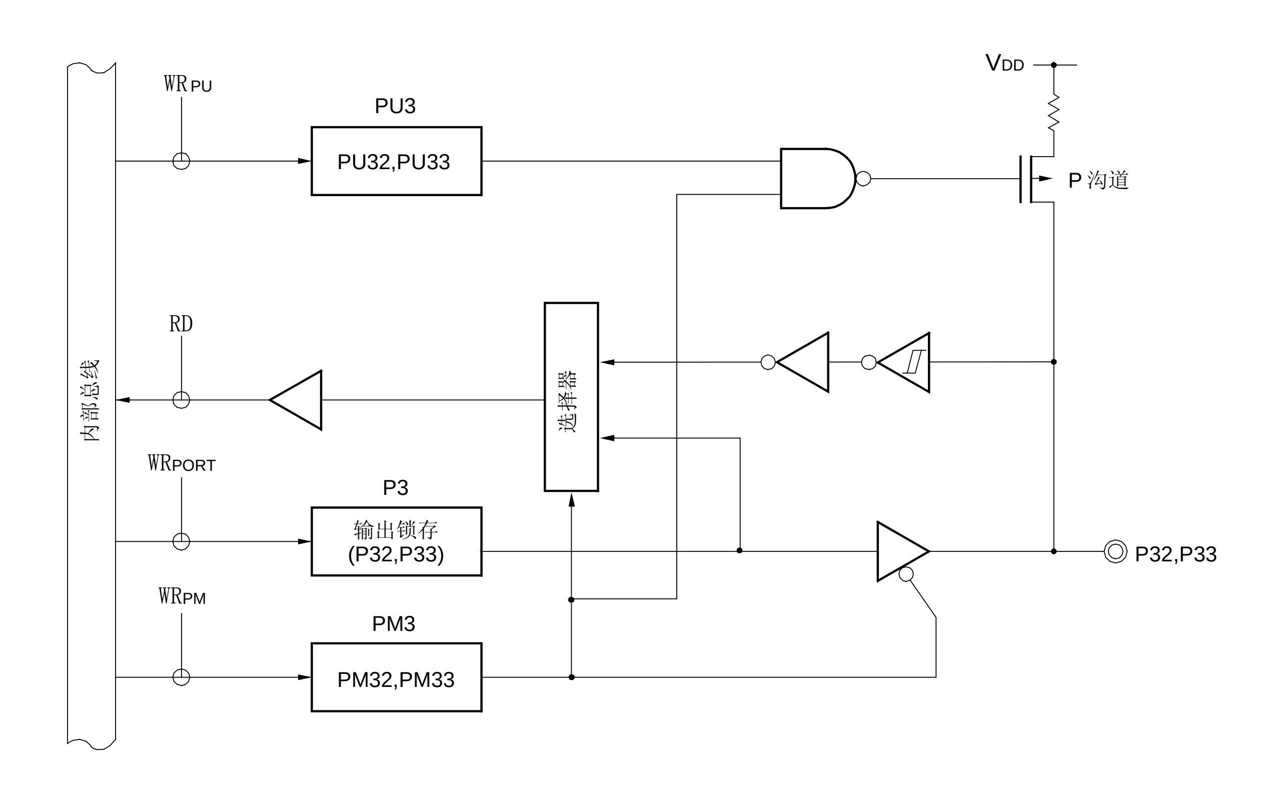
PU3：　上拉电阻选择寄存器 3
P3：　端口寄存器 3
PM3：　端口模式寄存器 3
RD：　读信号
WR××：　写信号

图 4-5.　P31 的框图



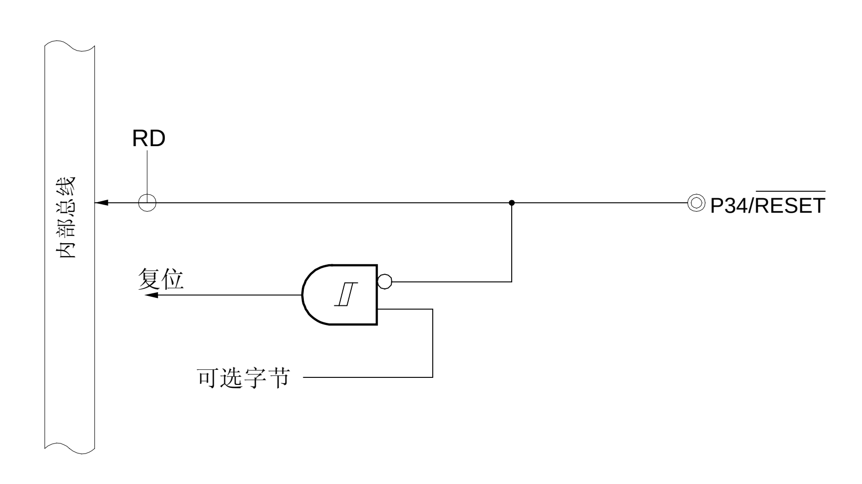
PU3：　上拉电阻选择寄存器 3
P3：　端口寄存器 3
PM3：　端口模式寄存器 3
RD：　读信号
WR××：　写信号

图 4-6.　P32 和 P33 的框图



PU3：　上拉电阻选择寄存器 3
P3：　端口寄存器 3
PM3：　端口模式寄存器 3
RD：　读信号
WR××：　写信号

图 4-7.　P34 的框图



RD：　读信号

注意事项　因为引脚 P34 可以作为 $\overline{\text{RESET}}$ 引脚使用，如果当它作为输入端口引脚使用的时候，从外部输入一个复位信号也不能实现复位功能。这个端口的功能通过可选字节来选择，具体细节清参照第十八章 可选字节。

如果复位状态被 POC 电路释放后，再次参照可选字节之前，一个低电平被输入到 $\overline{\text{RESET}}$ 引脚，78K0S/KB1+被复位并保持在复位状态直到一个高电平输入到 $\overline{\text{RESET}}$ 引脚。

4.2.4 端口 4

端口 4 是一个带输出锁存的 8 位 I/O 端口。端口的每一位都可以通过端口模式寄存器 4(PM4)设置成输入或者输出模式。当引脚 P40 至 P47 被当作输入端口使用时，可以以 1 位为单位设置上拉电阻选择寄存器 4(PU4)来连接片内上拉电阻。

P41 至 P44 有输出定时器信号，输入外部中断请求信号和输入/输出串行接口的数据的功能。

复位后端口 4 为输入模式。

图 4-8 至 4-11 显示了端口 4 的框图。

图 4-8.　P40，P45 至 P47 的框图

PU4：　上拉电阻选择寄存器 4

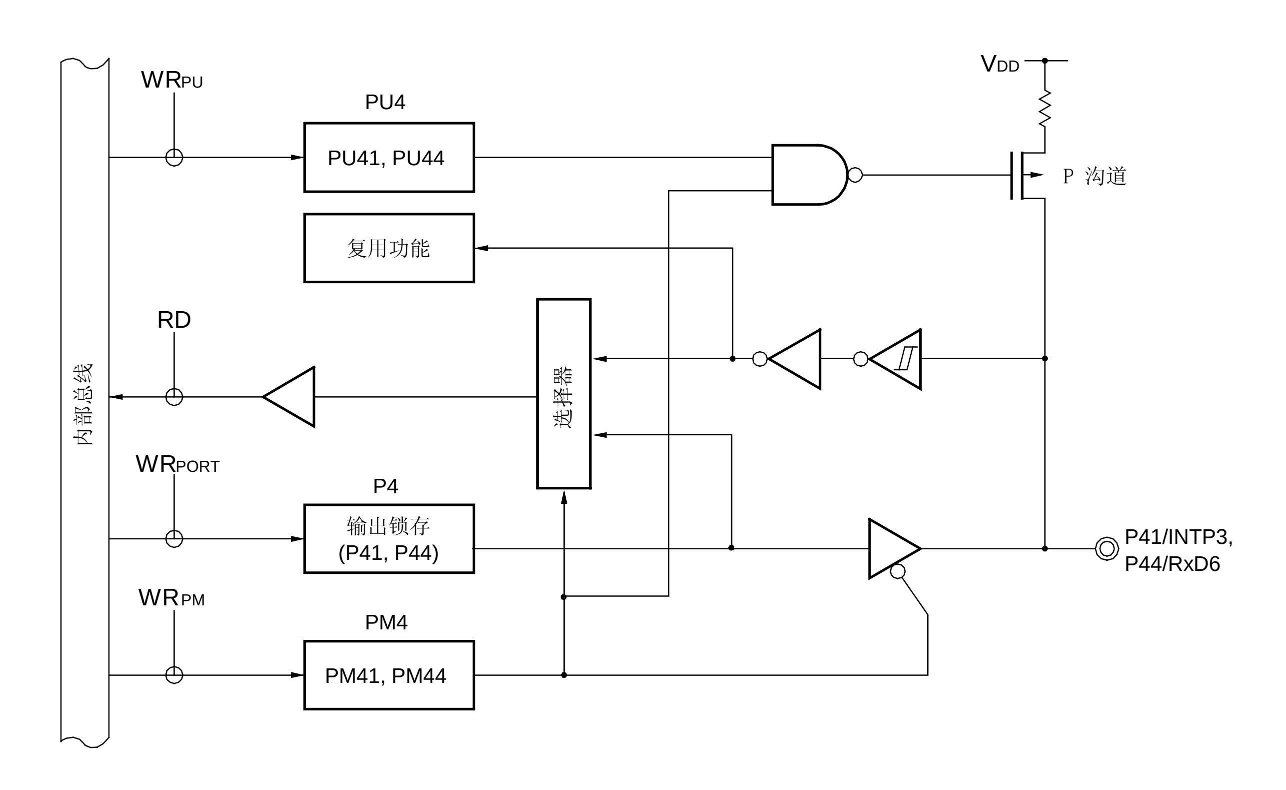
P4：　端口寄存器 4

PM4：　端口模式寄存器 4

RD：　读信号

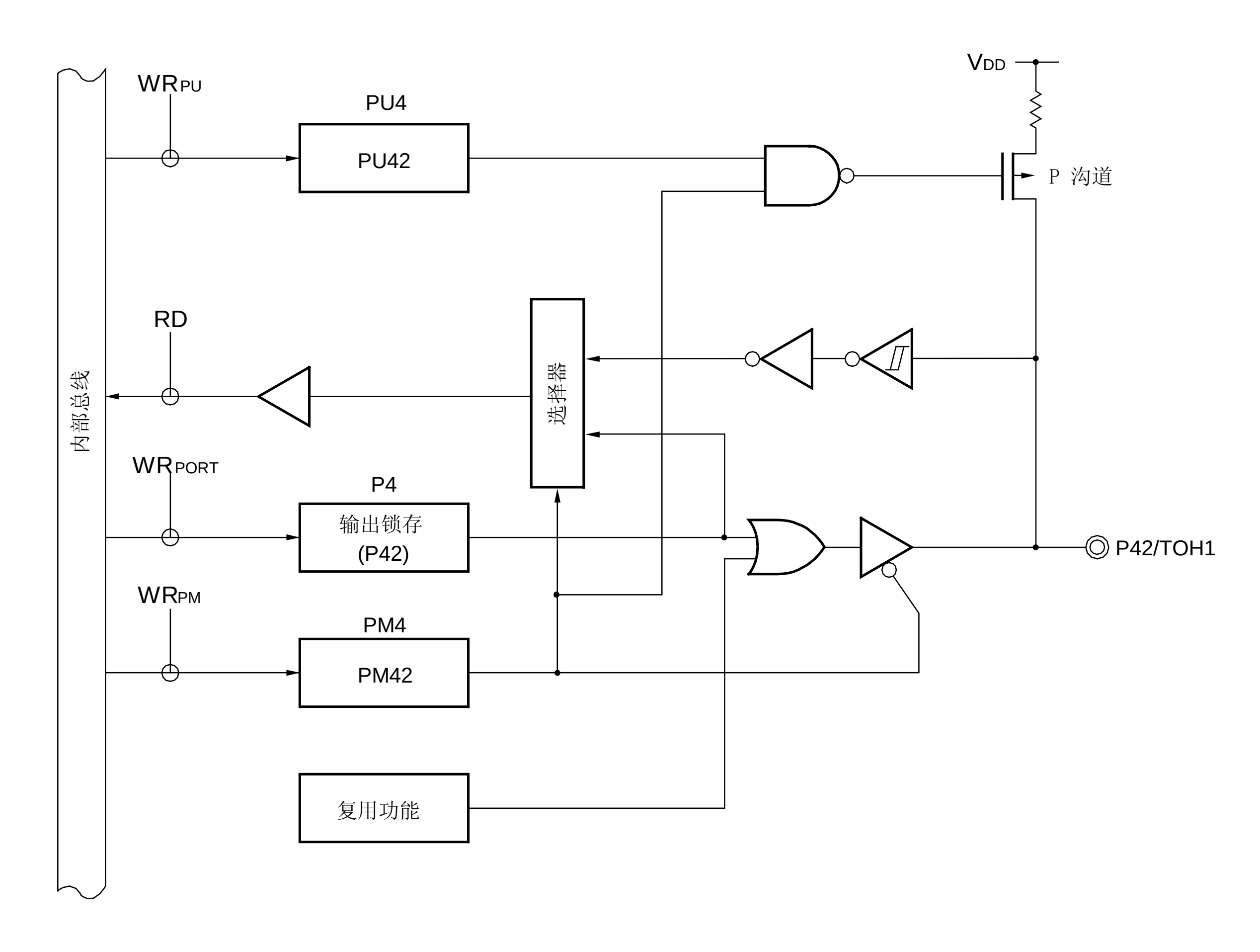
WR××：　写信号

图 4-9.　P41 和 P44 的框图



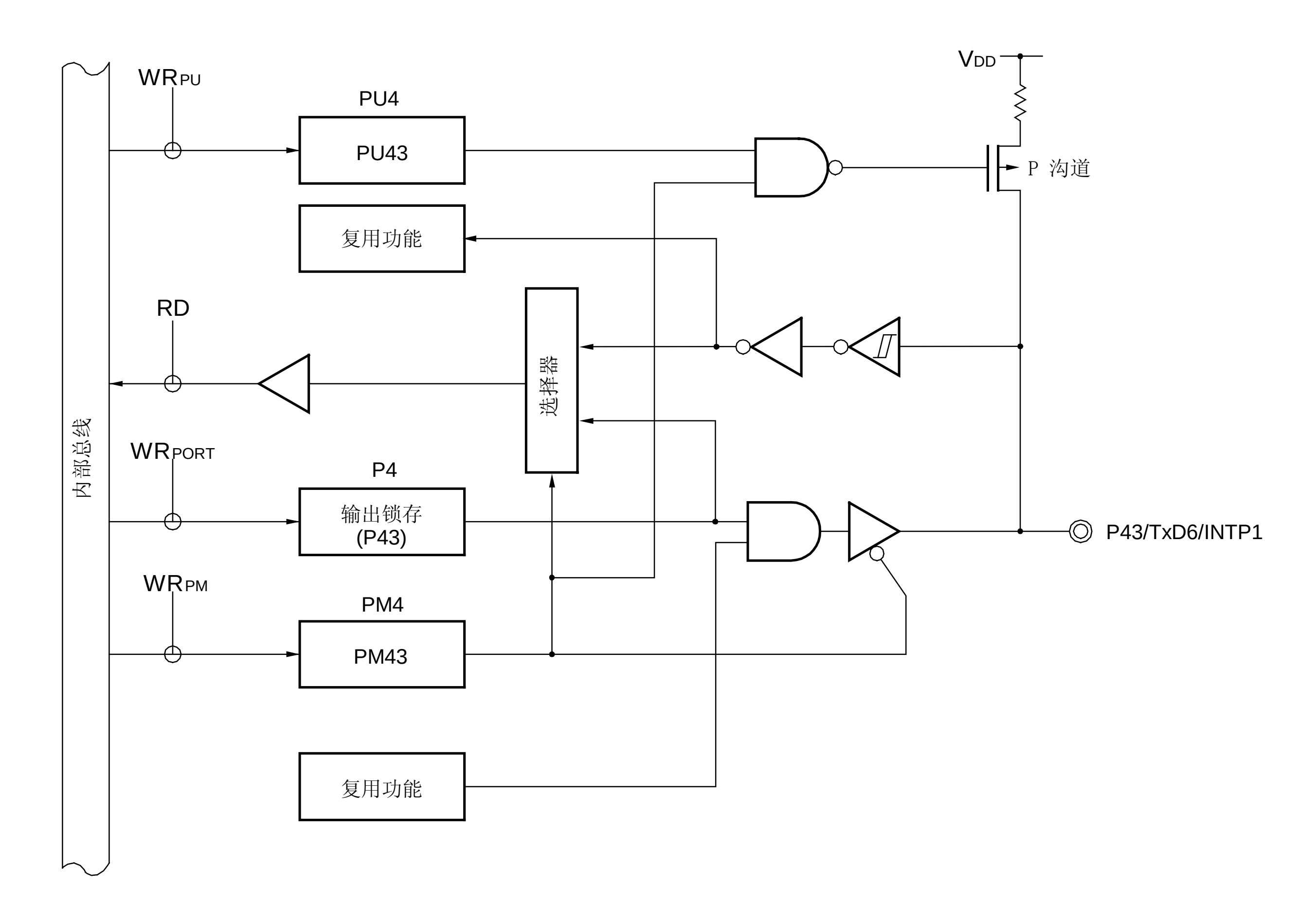
PU4：　上拉电阻选择寄存器 4
P4：　端口寄存器 4
PM4：　端口模式寄存器 4
RD：　读信号
WR××：　写信号

图 4-10.　P42 的框图



PU4：　上拉电阻选择寄存器 4
P4：　端口寄存器 4
PM4：　端口模式寄存器 4
RD：　读信号
WR××：　写信号

图 4-11.　P43 的框图



PU4:　上拉电阻选择寄存器 4
P4:　端口寄存器 4
PM4:　端口模式寄存器 4
RD:　读信号
WR××:　写信号

4.2.5 端口 12

端口 12 是一个带输出锁存的 4 位 I/O 端口。该端口可通过端口模式寄存器 12（PM12）位选输入或输出模式。当用作输入端口时，可通过上拉电阻选择寄存器 12（PU12）设置 P120 和 P123 使用内部上拉电阻。

P121 和 P122 引脚也能复用为系统时钟振荡器的 X1 和 X2 引脚。P121 和 P122 引脚的功能取决于系统时钟振荡器的选择。能够选择以下三种系统时钟振荡器。

(1) 高速内部振荡器

P121 和 P122 引脚能够被用作 I/O 端口引脚。

(2) 晶体/陶瓷振荡器

P121 和 P122 引脚作为 X1 和 X2 引脚使用时不能作为 I/O 端口引脚使用。

(3) 外部时钟输入

P121 引脚作为输入外部时钟的 X1 引脚时不能再作为 I/O 端口引脚使用。

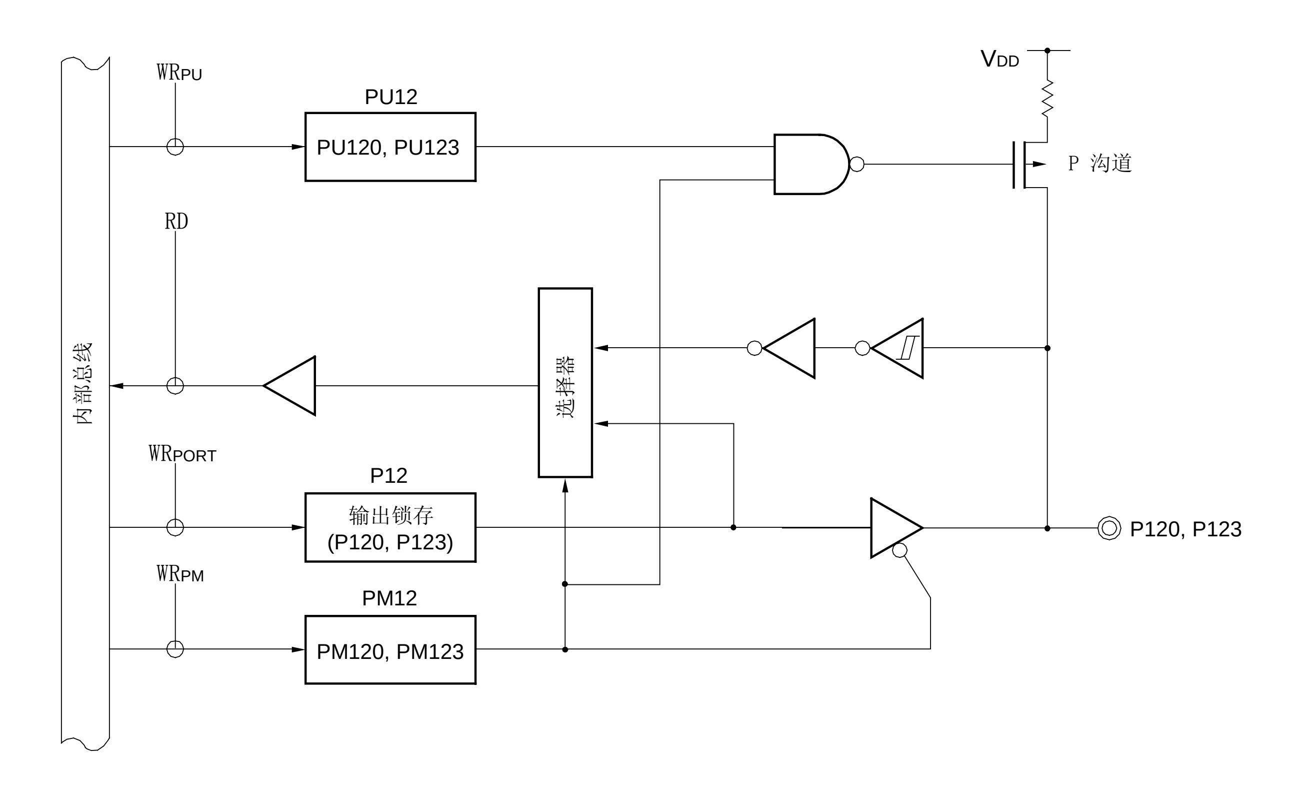
P122 引脚能被用作 I/O 端口引脚。

可通过可选字节选择系统时钟振荡器，详细描述请参见**第十八章 可选字节**。

复位后端口 12 为输入模式。

图 4-12 和 4-13 显示端口 12 的框图。

图 4-12. P120 和 P123 的框图



PU12: 上拉电阻选择寄存器 12

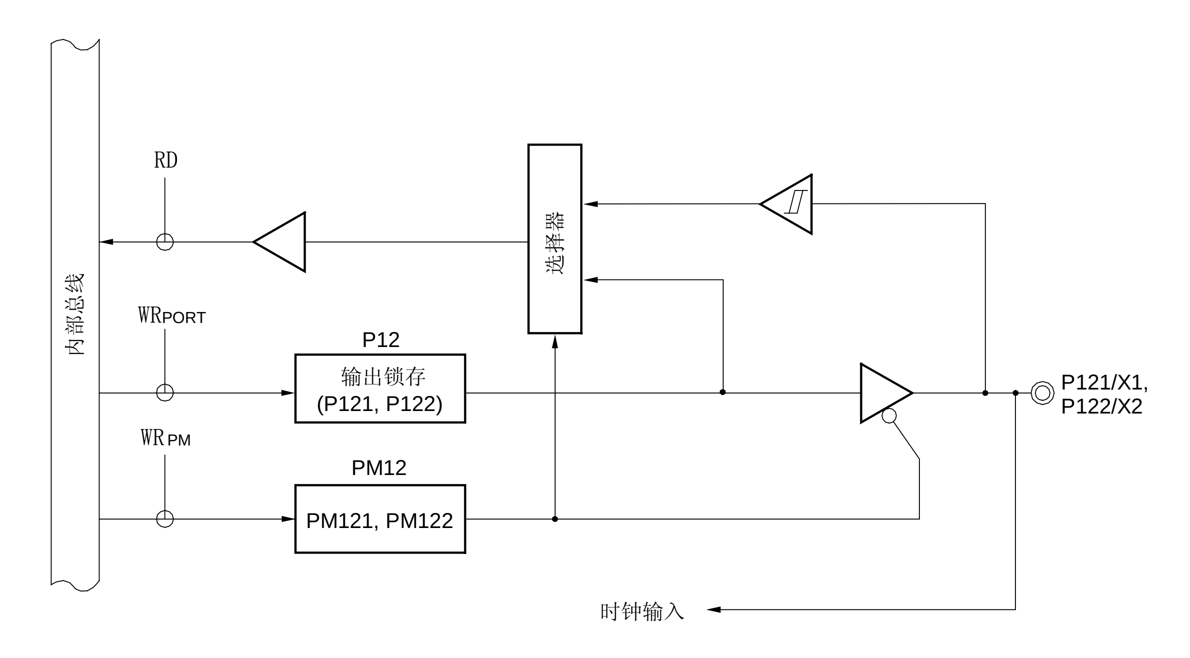
P12: 端口寄存器 12

PM12: 端口模式寄存器 12

RD: 读信号

WR××: 写信号

图 4-13.　P121 和 P122 的框图



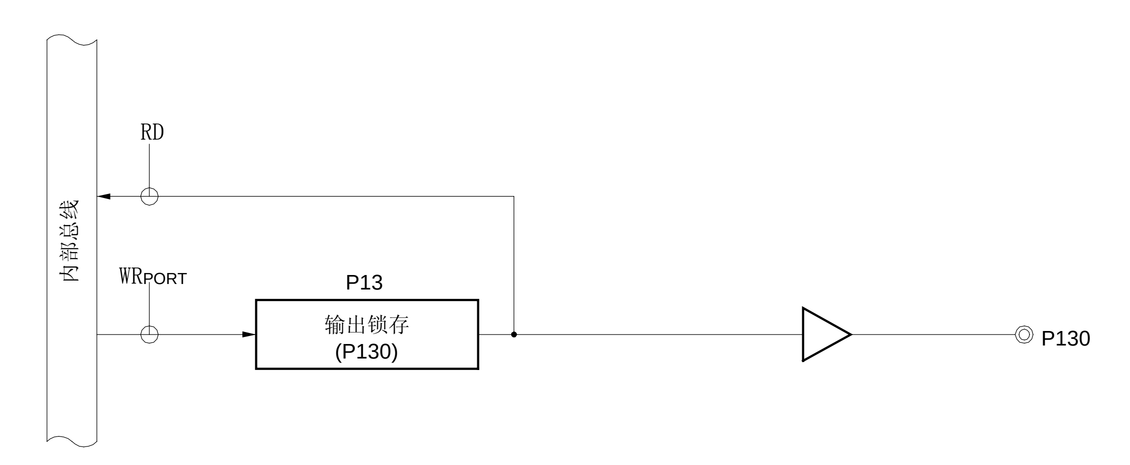
PM12：　端口模式寄存器 12
P12：　端口寄存器 12
RD：　读信号
WR××：　写信号

4.2.6 端口 13

这是一个 1 位输出端口。

图 4-14 显示了端口 13 的框图。

图 4-14.　P130 的框图



P13：　端口寄存器 13
RD：　读信号
WR××：　写信号

备注　当输入复位信号时，P130 引脚输出低电平。如果 P130 引脚在复位释放后立即输出一个高电平，则这个 P130 引脚的输出信号能作为一个虚拟的 CPU 复位信号。

4.3　控制端口功能的寄存器

端口被下列四种类型的寄存器所控制。

- 端口模式寄存器（PM0，PM2，PM3，PM4，PM12）
- 端口寄存器（P0，P2，P3，P4，P12，P13）
- 端口模式控制寄存器 2（PMC2）
- 上拉电阻选择寄存器（PU0，PU2，PU3，PU4，PU12）

(1) 端口模式寄存器 (PM0, PM2, PM3, PM4, PM12)

这类寄存器定义端口的 1 位输入或者输出模式。

可以由 1 位或者 8 位的操作指令设置寄存器。

复位后这些寄存器的值为 FFH。

如果端口引脚使用复用功能，可如表 4-3 所示设置端口模式寄存器和输出锁存。

注意事项 因为 P30, P31 和 P43 引脚也被用作外部中断引脚，这些引脚的任何一个被设置成输出模式并它的输出电平被改变，则相应的中断请求标志被置位。因此使用这个端口为输出模式时，预先设置相应的中断屏蔽标志为 1。

图 4-15. 端口模式寄存器的格式

地址： FF20H 复位后： FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

地址： FF22H 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

地址： FF23H 复位后： FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

地址： FF24H 复位后： FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

地址： FF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	PM123	PM122	PM121	PM120

PMmn	Pmn 引脚 I/O 模式的选择 (m = 0，2，3，4 或 12；n = 0～7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

(2)　端口寄存器（P0，P2，P3，P4，P12，P13）

这些寄存器用于设置从相应端口引脚向连接在芯片上的外部器件输出的数据。

当在读取端口寄存器时,在输入模式下将读取引脚的电平，在输出模式下将读取端口输出锁存器的值。

P00 至 P03，P20 至 P23，P30 至 P34，P40 至 P47，P120 至 P123 和 P130 可由 1 位或者 8 位存储器操作指令设置。

复位后这些寄存器的值为 00H。

图 4-16.　端口寄存器的格式

地址：FF00H　复位后：00H（输出锁存）　R/W

符号	7	6	5	4	3	2	1	0
P0	0	0	0	0	P03	P02	P01	P00

地址：FF02H　复位后：00H（输出锁存）　R/W

符号	7	6	5	4	3	2	1	0
P2	0	0	0	0	P23	P22	P21	P20

地址：FF03H　复位后：00H [注]（输出锁存）　R/W [注]

符号	7	6	5	4	3	2	1	0
P3	0	0	0	P34	P33	P32	P31	P30

地址：FF04H　复位后：00H（输出锁存）　R/W

符号	7	6	5	4	3	2	1	0
P4	P47	P46	P45	P44	P43	P42	P41	P40

地址：FF0CH　复位后：00H（输出锁存）　R/W

符号	7	6	5	4	3	2	1	0
P12	0	0	0	0	P123	P122	P121	P120

地址：FF0DH　复位后：00H（输出锁存）　R/W

符号	7	6	5	4	3	2	1	0
P13	0	0	0	0	0	0	0	P130

Pmn	m = 0，2，3，4，12 或 13；　n = 0～7	
	输出数据控制（在输出模式下）	输入数据读取（在输入模式下）
0	输出 0	输入低电平
1	输出 1	输入高电平

注　因为 P34 是只读的，所以它复位后的值不确定。

(3)　端口模式控制寄存器 2 (PMC2)

这个寄存器指定了端口模式或 A/D 转换模式。

PMC2 寄存器的每一位对应于端口 2 的每一个引脚并且可用 1 位控制。

PMC2 可以由 1 位或者 8 位的存储器操作指令设置。

复位后 PMC2 的值为 00H。

图 4-17.　端口模式控制寄存器 2 的格式

地址: FF84H　复位后: 00H　R/W

符号	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	工作模式选择 (n = 0 ～ 3)
0	端口模式
1	A/D 转换模式

表 4-3.　使用复用功能的时候，端口模式寄存器，端口寄存器(输出锁存)和端口模式控制寄存器的设置

引脚名称	复用功能引脚		PM××	P××	PMC2n
	名称	I/O			(n = 0 ～ 3)
P20 ～ P23	ANI0 ～ ANI3	输入	1	×	1
P30	TI000	输入	1	×	–
	INTP0	输入	1	×	–
P31	TO00	输出	0	0	–
	TI010	输入	1	×	–
	INTP2	输入	1	×	–
P41	INTP3	输入	1	×	–
P42	TOH1	输出	0	0	–
P43	TxD6	输出	0	1	–
	INTP1	输入	1	×	–
P44	RxD6	输入	1	×	–

备注　×　：　不考虑

PM××　：　端口模式寄存器，P××：端口寄存器 (端口具有输出锁存)

PMC2×　：　端口模式控制寄存器

(4) 上拉电阻选择寄存器（PU0，PU2，PU3，PU4 和 PU12）

这些寄存器指定是否使用引脚 P00 至 P03，P20 至 P23，P30 至 P33，P40 至 P47，P120 和 P123 的内置上拉电阻。
可以通过设置 PU0，PU2，PU3，PU4 或 PU12 来定义是否使用每个端口的上拉电阻。
PU0，PU2，PU3，PU4 和 PU12 可以由 1 位或者 8 位存储器操作指令设置。
复位后这些寄存器的值为 00H。

图 4-18. 上拉电阻选择寄存器的格式

地址：FF30H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PU0	0	0	0	0	PU03	PU02	PU01	PU00

地址：FF32H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PU2	0	0	0	0	PU23	PU22	PU21	PU20

地址：FF33H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PU3	0	0	0	0	PU33	P32	PU31	PU30

地址：FF34H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

地址：FF3CH 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
PU12	0	0	0	0	PU123	0	0	PU120

PUmn	Pmn引脚的上拉电阻寄存器的选择（m = 0，2，3，4或12；n = 0 ～ 7）
0	不连接上拉电阻
1	连接上拉电阻

4.4　端口功能操作

输入模式和输出模式的端口操作是不同的，具体如下所示。

注意事项　虽然 1 位存储器操作指令只对 1 位进行操作，但实际是以 8 位形式访问端口。因此，如果一个端口既可输入，又可输出，则对于端口中定义为输入模式的引脚的输出锁存器中的内容是不确定的。

4.4.1　写入 I/O 端口

(1)　**输出模式**

通过传送指令可以把数据写入到输出锁存器中。输出锁存器中的数据通过引脚输出。在新的数据被写入到输出锁存器中以前，输出锁存器中的数据不会改变。

复位后清空输出锁存器中的数据。

(2)　**输入模式**

通过传送指令可以把数据写入到输出锁存器中。因为输出缓冲已经关闭，引脚状态将不再被改变。

一旦数据被写入到输出锁存器当中，在新的数据被写入到输出锁存器中以前，输出锁存器中的数据不会改变。

复位后清空输出锁存器中的数据。

4.4.2　从 I/O 端口读入

(1)　**输出模式**

可以通过操作指令读取输出锁存器中的数据，输出锁存器中的数据不会被改变。

(2)　**输入模式**

可以通过操作指令读取引脚状态，输出锁存器中的内容不会被改变。

4.4.3　I/O 端口的操作

(1)　**输出模式**

对输出锁存器的内容执行一个操作，即对输出锁存器进行写操作。输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据写入。

复位后清空输出锁存器中的数据。

(2)　**输入模式**

读取引脚电平，对他的内容执行一个操作，则将引脚内容写入输出锁存器，但是如果输出缓冲已经关闭，则引脚状态将不再被改变。

复位后清空输出锁存器中的数据。

第五章 时钟发生器

5.1 时钟发生器的功能

时钟发生器包括两部分电路，一是供给 CPU 和外围硬件的时钟（系统时钟）产生电路，二是供给看门狗定时器和 8 位定时器 H1（TMH1）的时钟（间隔时间发生时钟）产生电路。

5.1.1 系统时钟振荡器

系统可以使用下列三种时钟振荡器。

- 高速 Ring-OSC 振荡器
  该电路可以产生 8MHz（典型值）的时钟信号，通过执行 STOP 指令使其振荡停止。
  如果使用高速 Ring-OSC 振荡器作为系统时钟，则 X1 和 X2 引脚可以作为输入/输出端口引脚使用。

- 晶体/陶瓷振荡器
  晶体/陶瓷振荡器产生的时钟信号，通过 X1 和 X2 引脚输入单片机。输入振荡频率的范围为 1～10MHz。通过执行 STOP 指令使其振荡停止。

- 外部时钟输入电路
  外部的集成电路产生的时钟信号，通过 X1 引脚输入单片机。输入振荡频率的范围为 1～10MHz。执行 STOP 指令后，单片机内部停止内部时钟信号。
  当外部时钟输入作为系统时钟信号时，X2 引脚可以作为输入/输出引脚使用。

通过设置可选字节可以指定系统时钟源。详细请参见 **第十八章 可选字节。**
如 X1 和 X2 引脚作为输入/输出端口引脚使用时，详细请参见 **第四章 端口功能**。

5.1.2 产生间隔时间的时钟振荡器

以下的电路用于产生间隔时间的时钟振荡器。

- 低速 Ring-OSC 振荡器
  低速 Ring-OSC 振荡器产生一个 240KHz（典型值）的时钟信号。通过对低速 Ring-OSC 模式寄存器（LSRCM）设置可以使振荡器停止工作。当设置可选字节时可以通过软件停止振荡器工作。

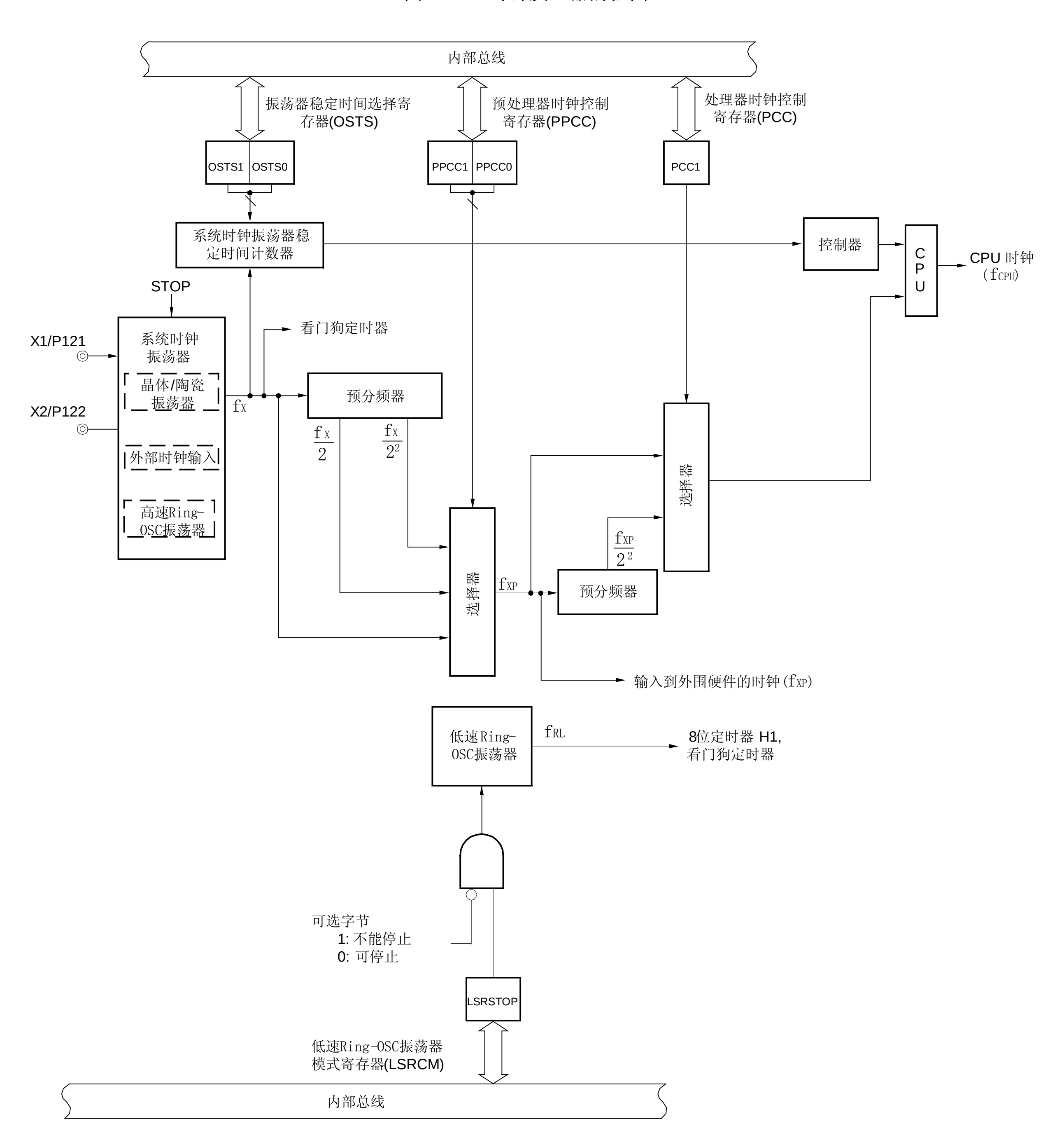
5.2 时钟发生器的配置

时钟发生器由下列硬件组成。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	处理器时钟控制寄存器（PCC） 预处理器时钟控制寄存器（PPCC） 低速 Ring-OSC 模式寄存器（LSRCM） 振荡稳定时间选择寄存器（OSTS）
振荡器	晶体/陶瓷振荡器 高速 Ring-OSC 振荡器 外部时钟输入电路 低速 Ring-OSC 振荡器

图 5-1. 时钟发生器的框图



注 由可选字节选择高速 Ring-OSC 振荡器、晶体/陶瓷振荡器或外部时钟输入作为系统时钟源。

5.3　控制时钟发生器的寄存器

下列四个寄存器控制时钟发生器。

- •处理器时钟控制寄存器(PCC)
- •预处理器时钟控制寄存器(PPCC)
- •低速 Ring-OSC 模式寄存器(LSRCM)
- •振荡稳定时间选择寄存器(OSTS)

(1) 处理器时钟控制寄存器（PCC）和预处理器时钟控制寄存器（PPCC）

这两个寄存器控制系统时钟的分频比。

PCC 和 PPCC 可以由 1 位或 8 位的存储器操作指令操作。

复位后，PCC 和 PPCC 的初始值为 02H。

图 5-2 处理器时钟控制寄存器（PCC）的格式

地址:FFFBH　复位后： 02H　R/W

符号	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	0	PCC1	0

注意事项　第 7 位到第 2 位和第 0 位必须设置为 0。

图 5-3 预处理器时钟控制寄存器（PPCC）的格式

地址:FFF3H　复位后： 02H　R/W

符号	7	6	5	4	3	2	1	0
PPCC	0	0	0	0	0	0	PPCC1	PPCC0

PPCC1	PPCC0	PCC1	选择 CPU 时钟 (f_{CPU})
0	0	0	F_X
0	1	0	$f_X/2$ 注 1
0	0	1	$f_X/2^2$
1	0	0	$f_X/2^2$ 注 2
0	1	1	$f_X/2^3$ 注 1
1	0	1	$f_X/2^4$ 注 2
其他状态			禁止设置

注　1. 当 PPCC=01H 时，供给外围设备的时钟 (f_{XP}) 为 $f_X/2$。

　　2. 当 PPCC=02H 时，供给外围设备的时钟(f_{XP})为 $f_X/2^2$。

78K0S/KB1+最快的指令可以在两个 CPU 时钟周期内完成。因此，CPU 时钟和指令最短执行时间的关系如下表 5-2 所示。

表 5-2 CPU 时钟和指令最短执行时间的关系

CPU 时钟（f_{CPU}）注	指令最短执行时间：$2/f_{CPU}$	
	高速 Ring-OSC 时钟（8.0 MHz（TYP.））	晶体/陶瓷振荡时钟或外部时钟输入（10.0 MHz）
f_X	0.25 μs	0.2 μs
$f_X/2$	0.5 μs	0.4 μs
$f_X/2^2$	1.0 μs	0.8 μs
$f_X/2^3$	2.0 μs	1.6 μs
$f_X/2^4$	4.0 μs	3.2 μs

注 CPU 时钟（高速 Ring-OSC 时钟、晶体/陶瓷振荡时钟或外部时钟输入）的选择通过可选字节来控制。

(2) 低速 **Ring-OSC** 模式寄存器(LSRCM)

这个寄存器用来选择低速 Ring-OSC 振荡器(240KHz（典型值）)的操作模式。

当通过可选字节设置低速 Ring-OSC 振荡器有效时，可通过软件使低速 Ring-OSC 振荡器停止工作。如果设置可选字节使低速 Ring-OSC 振荡器不能通过软件来停止工作，这种寄存器的设置是无效的，低速 Ring-OSC 振荡器继续工作。并且，WDT 的时钟源是由低速 Ring-OSC 振荡器给出的，详细请参见 **第九章 看门狗定时器**。

低速 Ring-OSC 模式寄存器可以由 1 位或 8 位存储器操作指令操作。

复位后，低速 Ring-OSC 模式寄存器的值为 00H。

图 5-4 低速 **Ring-OSC** 模式寄存器(LSRCM)的格式

地址：FF58H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	<0>
LSRCM	0	0	0	0	0	0	0	LSRSTOP

LSRSTOP	低速Ring-OSC振荡/停止状态
0	低速Ring-OSC工作
1	低速Ring-OSC停止

(3) 振荡稳定时间选择寄存器 (OSTS)

这个寄存器用于选择 STOP 模式释放时振荡器开始振荡的振荡稳定时间。仅当晶体/陶瓷振荡器时钟被选为系统时钟源并且在 STOP 模式释放后，通过 OSTS 设置的等待时间是有效的。如果是高速 Ring-OSC 振荡器或外部时钟输入作为系统时钟时，不需要振荡稳定时间。

上电后或复位模式释放后的系统时钟振荡器和振荡稳定时间的选择由可选字节决定。详情请参见 **第十八章 可选字节**。

OSTS 可以由 8 位存储器操作指令操作。

图 5-5 振荡稳定时间选择寄存器（OSTS）的格式

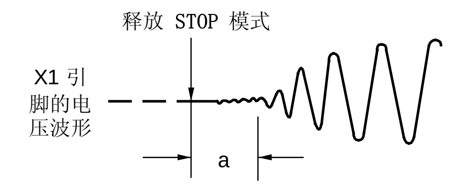
地址: FFF4H 复位后：不确定 R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	0	OSTS1	OSTS0

OSTS1	OSTS0	振荡稳定时间的选择
0	0	2^{10}/fx (102.4 μs)
0	1	2^{12}/fx (409.6 μs)
1	0	2^{15}/fx (3.27 ms)
1	1	2^{17}/fx (13.1 ms)

注意事项 1. 设置和释放 STOP 模式时按照下面的方法设置振荡稳定时间：OSTS 设置的振荡稳定时间大于等于预期的振荡器振荡稳定时间。

2. 无论 STOP 模式是被复位信号打破还是被中断打破，STOP 模式释放后的等待时间不包括从释放 STOP 模式到晶振开始震荡的时间（下图中的 a）。



注意事项 3.上电或者复位后的振荡稳定时间可以通过可选字节来设置，详情参见 第十八章 可选字节。

备注 1. ( )：fx = 10 MHz

2. 可通过查看所使用的振荡器的特性决定振荡器的振荡稳定时间。

5.4 系统时钟振荡器

系统时钟振荡器可以由下列三种振荡器构成：

- 高速 Ring-OSC 振荡器，工作在 8MHz（典型值）。
- 晶体/陶瓷振荡器，工作在 1～10MHz。
- 外部时钟输入电路，由 X1 引脚输入 1～10MHz 时钟信号。

5.4.1 高速 Ring-OSC 振荡器

78K0S/KB1+包含一个高速 Ring-OSC 振荡器（8MHz（典型值））。

如果可选字节设置高速 Ring-OSC 作为系统时钟源，则 X1 和 X2 引脚可以作为输入/输出端口引脚使用。

关于可选字节的详细情况请参见 **第十八章 可选字节**。

关于输入/输出端口的详细情况，请参见 **第四章 端口功能**。

5.4.2 晶体/陶瓷振荡器

晶体/陶瓷振荡器通过与 X1 和 X2 引脚相连的晶体或陶瓷振荡器产生振荡时钟信号。

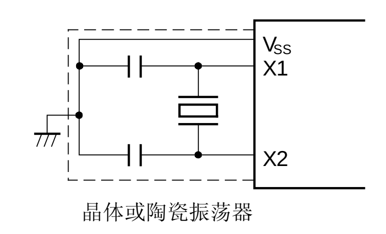
如果通过可选字节设置晶体/陶瓷振荡器作为系统时钟源，则 X1 和 X2 引脚是晶体或陶瓷振荡器的连接引脚。

关于可选字节的详细情况请参见 **第十八章 可选字节**。

关于输入/输出端口的详细情况，请参见 **第四章 端口功能**。

图 5-6 显示了晶体/陶瓷振荡器的外部电路。

图 5-6 晶体/陶瓷振荡器的外部电路



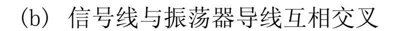
注意事项 **当使用晶体/陶瓷振荡器时，布线应该按照图 5-6 中虚线内的方式连接，这样可以减小分布电容对振荡器的影响。**

- **保持连线尽量的短。**
- **不要使其它的信号线与振荡器导线互相交叉，不要在振荡器旁边通过大电流的信号线。**
- **保持振荡器电容的地和 Vss 同电位，不要让电容使用的地通过大电流。**
- **不要从振荡器上取信号。**

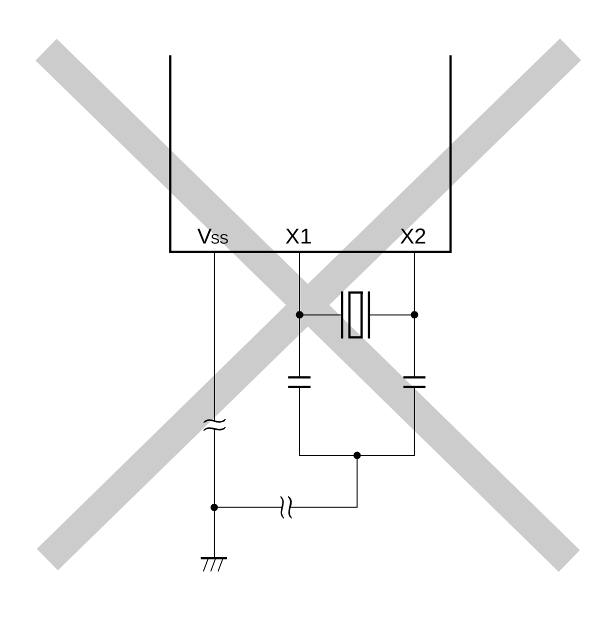
图 5-7 显示了振荡器错误连接方式的示例。

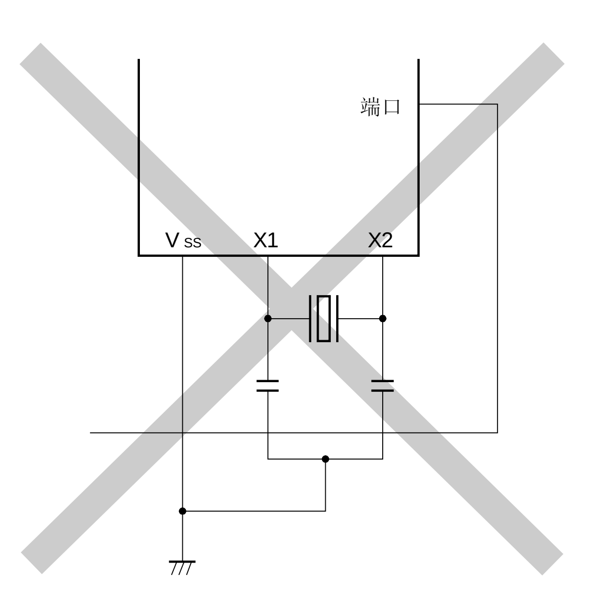
图 5-7 振荡器错误连接方式的示例 (1/2)

(a) 连接线太长



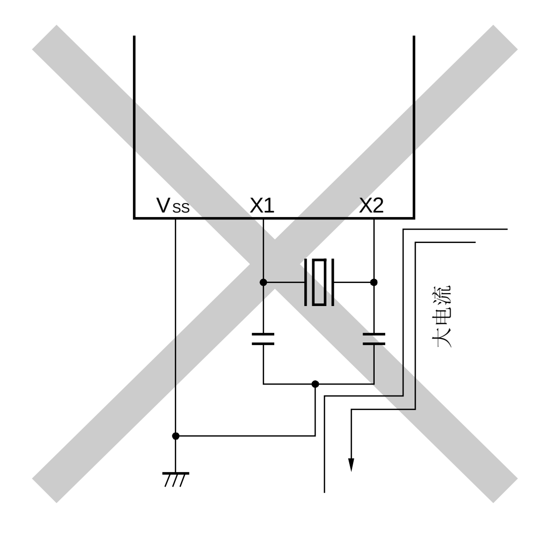
(b) 信号线与振荡器导线互相交叉





(c) 附近流过大的电流

(d) 大电流通过振荡器的地线
(A, B 和 C 电位波动)



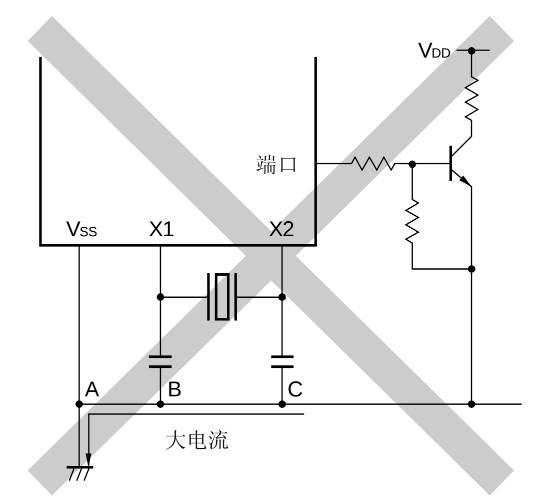
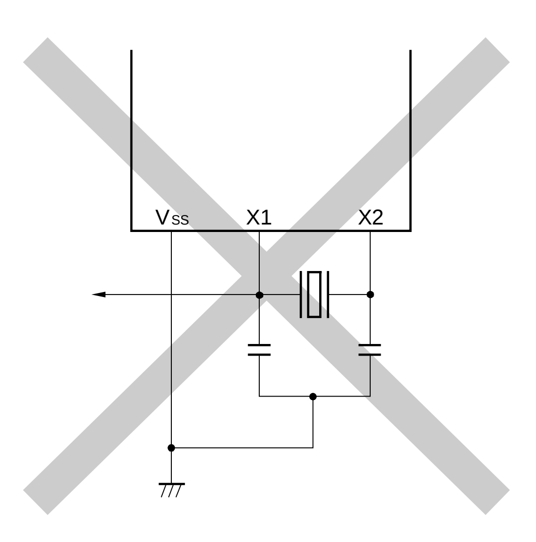


图 5-7 振荡器错误连接方式的示例 (2/2)

(e) 从振荡器取信号



5.4.3 外部时钟输入电路

该电路通过 X1 引脚从外部 IC 输入时钟信号。

如果通过可选字节选择外部时钟输入信号作为系统的时钟源，则 X2 引脚可以作为输入/输出端口引脚使用。

关于可选字节的详细情况请参见 **第十八章 可选字节**。

关于输入/输出端口的详细情况，请参加 **第四章 端口功能**。

5.4.4 预分频器

预分频器将系统时钟振荡器的时钟（fx）输出分频后产生一个时钟（f_{XP}）给外围硬件，也能将外围硬件的时钟（f_{XP}）分频后作为 CPU 时钟。

备注 时钟输出是由可选字节选择的振荡器（高速 Ring-OSC 振荡器，晶体/陶瓷振荡器或外部时钟输入电路）分频得到。关于可选字节的详细情况请参见 **第十八章 可选字节**。

5.5 CPU 时钟发生器的操作

CPU 的时钟信号(f_{CPU})来自于下列三种振荡器中的一种产生的系统时钟：

•高速 Ring-OSC 振荡器，工作在 8MHz（典型值）。
•晶体/陶瓷振荡器，工作在 1～10MHz。
•外部时钟输入电路，由 X1 引脚输入 1～10MHz 时钟信号。

由可选字节选择系统时钟振荡器。关于可选字节的详情请参照 **第十八章 可选字节**。

(1) 高速 Ring-OSC 振荡器

当通过可选字节选择高速 Ring-OSC 振荡器时，情况如下：

- 较短的启动时间
  如果选择高速 Ring-OSC 振荡器作为振荡器，则 CPU 不需要等待系统时钟的振荡稳定时间就可以开始工作。因此，启动时间缩短了。
- 扩展性提高
  如果选择高速 Ring-OSC 振荡器作为振荡器，则 X1 和 X2 引脚就可以作为输入/输出端口引脚使用。
  详情请参见 **第四章 端口功能**。

默认由高速 Ring-OSC 振荡器启动时的时序图和状态转换框图如下图 5-8，5-9 所示。

备注 当使用高速 Ring-OSC 振荡器时，时钟精确度为 ±5%。

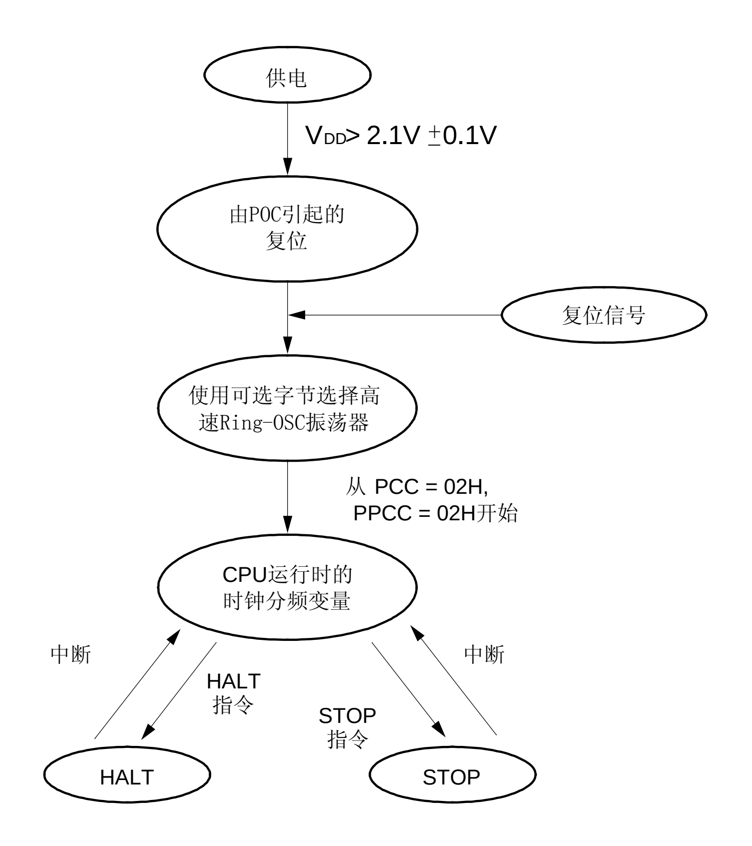
图 5-8 默认由高速 Ring-OSC 振荡器启动时的时序图

注 操作停止时间为：277μs（最小值），544μs（典型值），1.075ms（最大值）。

(a) 通过上电时的上电清零功能产生内部复位信号，复位后参照可选字节，选择系统时钟。

(b)参照可选字节，选择系统时钟。然后，高速 Ring-OSC 时钟作为系统时钟工作。

图 5-9 默认由高速 **Ring-OSC** 振荡器启动时的状态转换框图



备注 PCC：处理器时钟控制寄存器

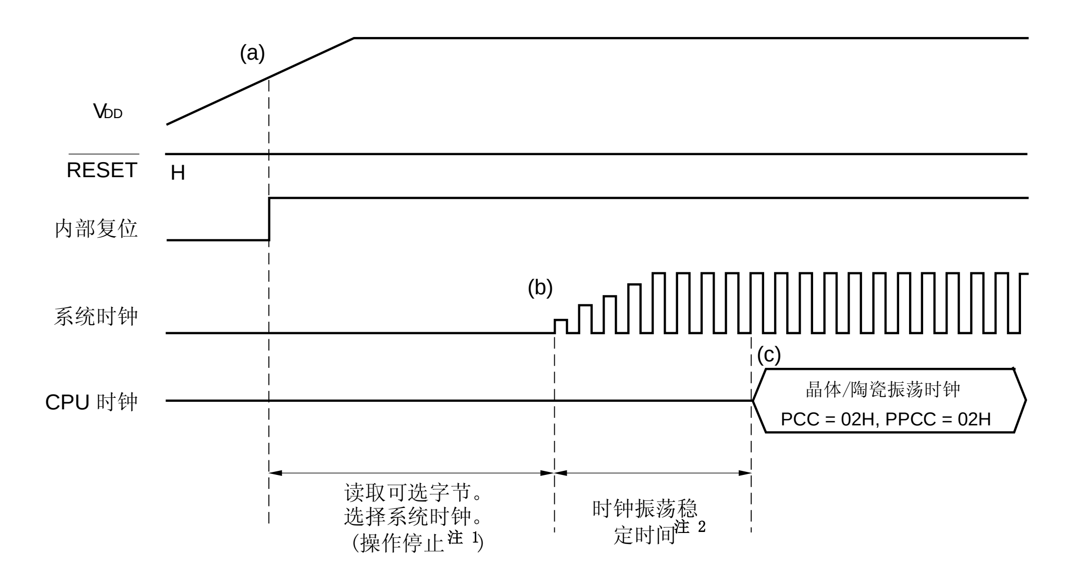
PPCC：预处理器时钟控制寄存器

(2) **晶体/陶瓷振荡器**

如果通过可选字节选择晶体/陶瓷振荡器，晶体/陶瓷振荡器的时钟频率可在 1～10MHz 之间选择，与高速 Ring-OSC 振荡器（8MHz（典型值））相比，因为频率偏差更小所以处理精度更高。

默认由晶体/陶瓷振荡器启动时的时序图和状态转换框图如下图 5-10，5-11 所示。

图 5-10 默认由晶体/陶瓷振荡器启动时的时序图



注 1. 操作停止时间为：276μs（最小值），544μs（典型值），1.074ms（最大值）。

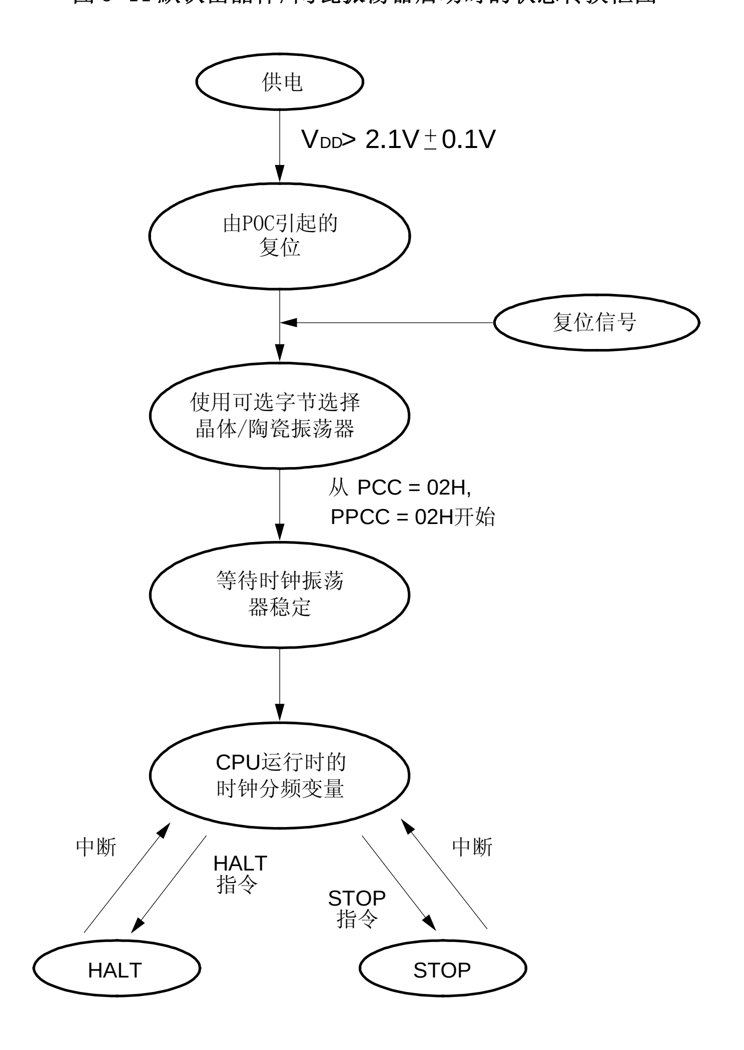
2. 默认启动时的时钟振荡稳定时间由可选字节设定。详情请参见 **第十八章 可选字节**。通过时钟振荡稳定时间寄存器（OSTS）选择复位模式释放后的时钟振荡稳定时间。

(a) 通过上电时的上电清零功能产生内部复位信号，复位后参照可选字节，选择系统时钟

(b) 在高速 Ring-OSC 时钟产生之后，参照可选字节，选择系统时钟。在这种情况下，选择晶体/陶瓷振荡器为系统时钟。

(c) 如果选择晶体/陶瓷振荡器为系统时钟，在时钟振荡稳定过后，它作为 CPU 时钟开始工作。等待时间通过可选字节选择。详情请参见 **第十八章 可选字节。**

图 5-11 默认由晶体/陶瓷振荡器启动时的状态转换框图



备注 PCC：处理器时钟控制寄存器
PPCC：预处理器时钟控制寄存器

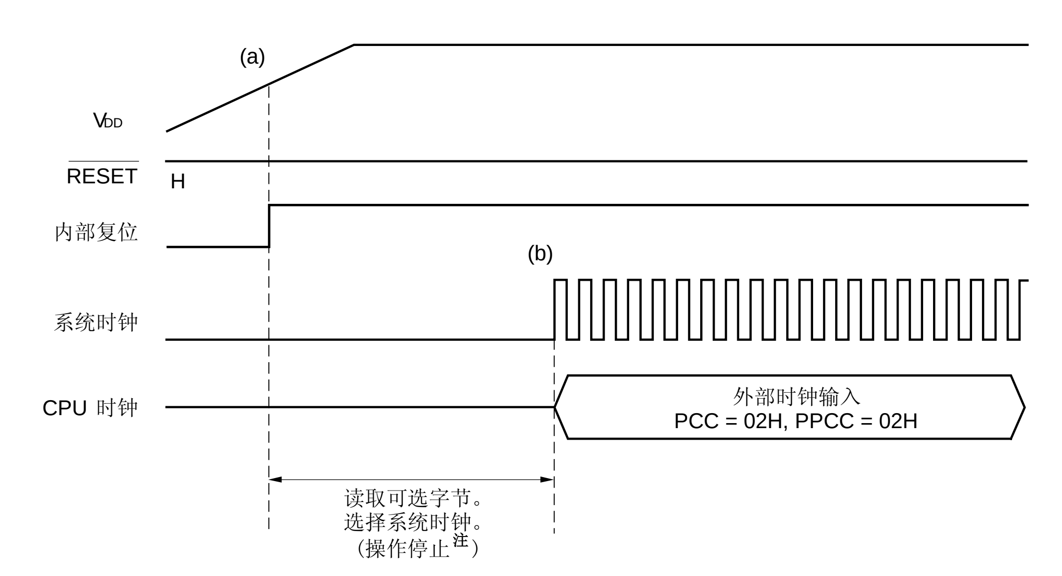
(3) **外部时钟输入电路**

当通过可选字节选择外部时钟输入时，情况如下：

- 高速操作
  因为外部时钟频率可在 1 至 10MHz 之间的选择，并且外部时钟的频率偏差较小，所以同高速 Ring-OSC 振荡器（8MHz（典型值））相比，外部时钟输入具有较高的精确度。
- 扩展性提高
  如果选择外部时钟输入作为振荡器，则 X2 引脚就可以作为输入/输出端口引脚使用。
  详情请参见 **第四章 端口功能**。

默认由外部时钟输入启动时的时序图和状态转换框图如下图 5-12，5-13 所示。

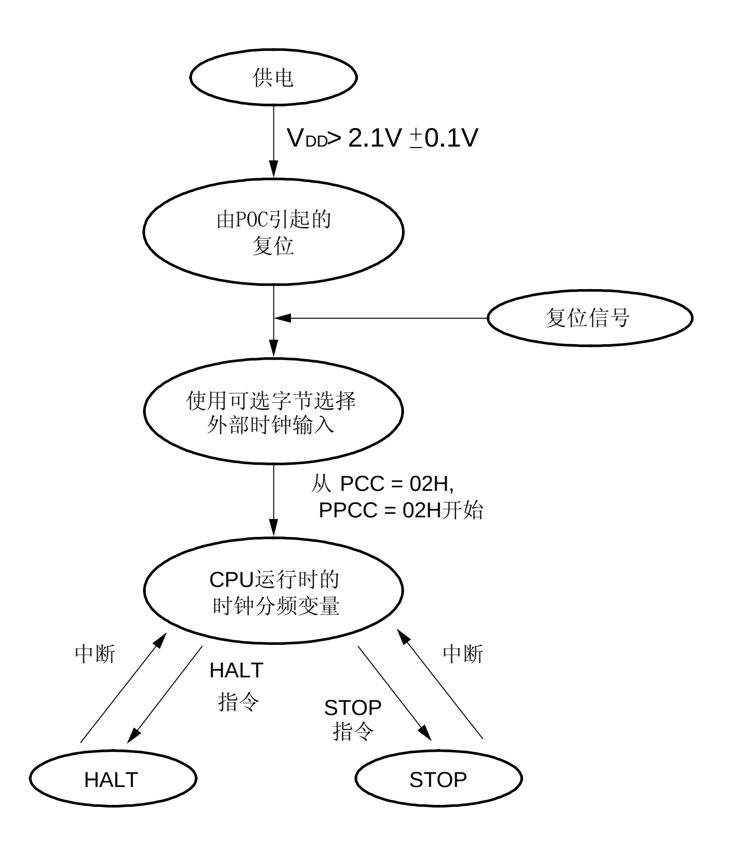
图 5-12 默认由外部时钟输入启动时的时序图



注　操作停止时间为：277μs（最小值），544μs（典型值），1.075ms（最大值）。

(a)　通过上电时的上电清零功能产生内部复位信号，复位后参照可选字节，选择系统时钟。
(b)　参照可选字节，选择系统时钟。外部时钟输入作为系统时钟工作。

图 5-13 默认由外部时钟输入启动时的状态转换框图



备注　PCC：处理器时钟控制寄存器
　　　PPCC：预处理器时钟控制寄存器

5.6 外围硬件时钟发生器的操作

下列两种时钟信号供给外围硬件使用：

- 外围硬件时钟信号（f_{XP}）
- 低速 Ring-OSC 时钟（f_{RL}）

(1) 外围硬件时钟信号

系统时钟（fx）分频得到外围硬件的时钟。分频比由预处理器时钟控制寄存器（PPCC）选择。
外围硬件时钟可以使用如下三种时钟频率：fx，fx/2，fx/4。表 5－3 列举了外围硬件时钟的配置情况。

表 5－3 外围硬件时钟信号

PPCC1	PPCC0	外围硬件时钟信号的选择（f_{XP}）
0	0	f_X
0	1	$f_X/2$
1	0	$f_X/2^2$
1	1	禁止

(2) 低速 Ring-OSC 时钟

在复位释放后，用作产生间隔时间时钟振荡器的低速 Ring-OSC 振荡器开始工作，振荡频率为 240 kHz （典型值）。
可由可选字节指定低速 Ring-OSC 振荡器是否能用软件停止。如果设置为通过软件可以停止低速 Ring-OSC 振荡器工作，则振荡器可以通过低速 Ring-OSC 振荡模式控制寄存器（LSRCM）控制启动或停止。 如果设定为不能用软件停止低速 Ring-OSC 振荡器，WDT 的时钟信号源只能是低速 Ring-OSC 时钟（f_{RL}）。
低速 Ring-OSC 振荡器时钟独立于 CPU 的时钟。因此，如果用它来作为 WDT 的时钟源，即使 CPU 时钟停止，也可检测到 CPU 挂起。如果把它用作为 8 位定时器 H1 的时钟源，则 8 位定时器 H1 可以在待机状态下工作。

表 5-4 显示了当低速 Ring-OSC 振荡器作为 WDT 和 8 位定时器 H1 时钟源时，低速 Ring-OSC 振荡器的工作状态。
图 5-14 显示了低速 Ring-OSC 振荡器的状态转换。

表 5－4 低速 Ring-OSC 振荡器的工作状态

可选字节设置		CPU状态	WDT状态	TMH1状态
不能通过软件设置停止工作	LSRSTOP ＝ 1	工作模式	停止	停止
	LSRSTOP ＝ 0		工作	工作
	LSRSTOP ＝ 1	待机	停止	停止
	LSRSTOP ＝ 0		停止	工作
不能停止工作		工作模式	工作	
		待机		

图 5-14 低速 **Ring-OSC** 振荡器的状态转换

供电

V_{DD}> 2.1V ±0.1V

由POC引起的复位

复位信号

通过可选字节选
择是否停止低速
Ring-Soc振荡器

可以停止

不能停止

通过软件[注] 选
择WDT的时钟源

低速Ring-Soc振荡
器可以停止

低速Ring-Soc振荡
器不能停止

WDT的时钟源
固定为f_{RL}

LSRSTOP = 1

LSRSTOP = 0

低速Ring-Soc振
荡器停止

注 看门狗(WDT)的时钟源可以为 f_X 或 f_{RL}，或者它可能停止。详情请参见 **第九章 看门狗定时器**。

第六章　16 位定时器/事件计数器 00

6.1　16 位定时器/事件计数器 00 的功能

16位定时器/事件计数器 00 有如下功能。

(1) 间隔定时器

16 位定时器/事件计数器 00 以预置时间间隔产生中断请求。

- 计数的范围: 2 ～ 65536

(2) 外部事件计数器

16 位定时器/事件计数器 00 可以测量外部输入的有效脉冲个数。

- 有效脉冲宽度为: 16/f_{XP} 或更大

(3) 脉冲宽度测量

16 位定时器/事件计数器 00 可以测量外部输入信号的脉冲宽度。

- 有效脉冲宽度为: 2/f_{XP} 或更大

(4) 方波输出

16 位定时器/事件计数器 00 可以输出任选频率的方波。

- 周期: (2 x 2 ～ 65536 × 2) × 计数时钟周期

(5) PPG 输出

16 位定时器/事件计数器 00 可以输出任意周期和脉冲宽度的方波。

- 1 < 脉冲宽度 < 周期 ≤ 65536

(6) 单脉冲输出

16 位定时器/事件计数器 00 可以输出任意脉冲宽度的单脉冲。

6.2　16 位定时器/事件计数器 00 的配置

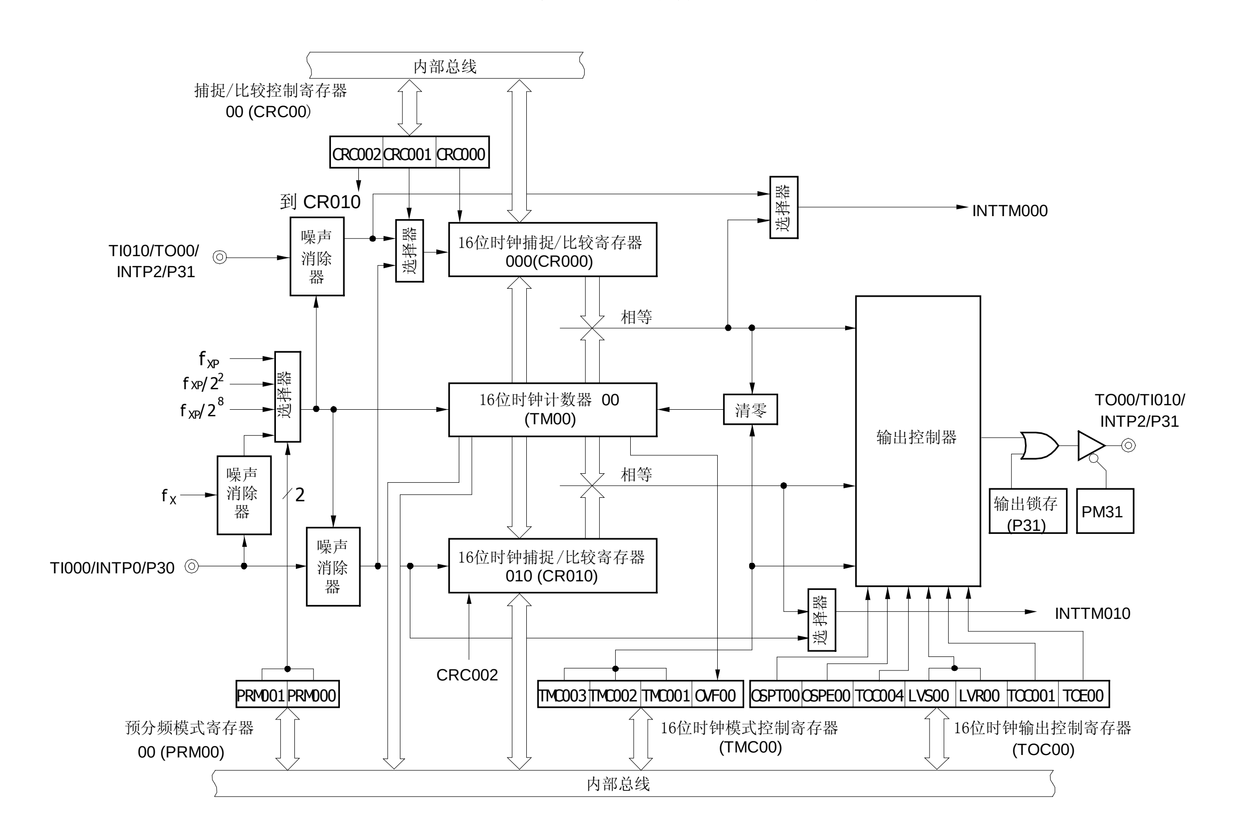
16 位定时器/事件计数器 00 由如下硬件组成。

表 6-1.　16 位定时器/事件计数器 00 的配置

项目	配置
定时器计数器	16 位定时器计数器 00 (TM00)
寄存器	16 位定时器捕捉/比较寄存器 000, 010 (CR000, CR010)
定时器输入	TI000，TI010
定时器输出	TO00，输出控制器
控制寄存器	16 位定时器模式控制寄存器 00 (TMC00) 捕捉/比较控制寄存器 00 (CRC00) 16 位定时器输出控制寄存器 00 (TOC00) 预分频模式寄存器 00 (PRM00) 端口模式寄存器 3 (PM3) 端口寄存器 3 (P3)

图 6-1 为这些计数器的框图。

图 6-1.　16 位定时器/事件计数器 00 框图

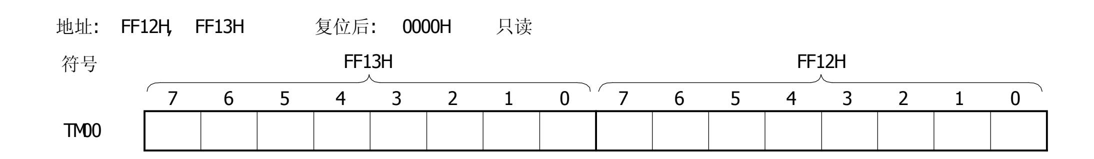


(1) 16 位定时器计数器 00 (TM00)

TM00 是 16 位只读寄存器，用于脉冲计数。

计数器随计数时钟的上升沿同步增加。如果在运行过程中读取计数器的值，那么计数时钟的输入将被暂停，并读取该时刻的计数值。

图 6-2. 16 位定时器计数器 00 (TM00)的格式



如下情况中计数值置为 0000H。

〈1〉 复位信号产生时
〈2〉 TMC003 和 TMC002 清零时
〈3〉 在清零&启动模式中，如果输入 TI000 的有效沿时
〈4〉 在清零&启动模式中，当 TM00 和 CR000 相等时
〈5〉 如果 OSPT00 在单脉冲输出模式下置 1

注意事项　1. TM00 被读取时，CR010 无法捕捉其值。
2. 当读 TM00 时，计数时钟不会停止。

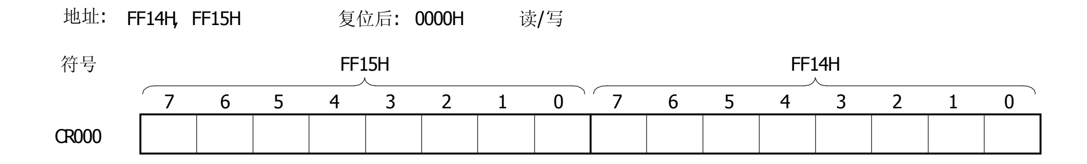
(2) 16 位定时器捕捉/比较寄存器 000 (CR000)

CR000 是 16 位寄存器，具有捕捉寄存器和比较寄存器的功能。无论是作为比较寄存器使用还是作为捕捉寄存器使用，均由捕捉/比较控制寄存器 00（CRC00）的第 0 位（CRC000）来设置。

CR000 由 16 位存储器操作指令设置。

复位信号产生后，CR000 清零。

图 6-3. 16 位定时器捕捉/比较寄存器 000 (CR000)的格式



- **当 CR000 作为比较寄存器使用**

  CR000 的设定值与 16 位定时器/计数器 00(TM00)的计数值不断比较，当值相等时产生中断请求(INTTM000)。当设置 TM00 作为间隔定时器时，也可以作为保存间隔时间的寄存器使用。

- **当 CR000 作为捕捉寄存器使用**

  可选择 TI000 或 TI010 引脚的有效沿作为捕捉触发。通过预分频模式寄存器 00(PRM00)设置 TI000 或 TI010 的有效沿。(参见表 6-2)

表 6-2. CR000 捕捉触发和 TI000 与 TI010 引脚的有效沿

(1) 选择 TI000 引脚有效沿作为捕捉触发 (CRC001 = 1, CRC000 = 1)

CR000 捕捉触发	TI000 引脚有效沿		
		ES010	ES000
下降沿	上升沿	0	1
上升沿	下降沿	0	0
无捕捉操作	上升和下降沿均为有效沿	1	1

(2) 选择 TI010 引脚有效沿作为捕捉触发 (CRC001 = 0, CRC000 = 1)

CR000 捕捉触发	TI010 引脚有效沿		
		ES110	ES100
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升和下降沿均触发	上升和下降沿均为有效沿	1	1

备注　1. 禁止设置 ES010, ES000 = 1, 0 和 ES110, ES100 = 1, 0 。
2. ES010, ES000:　预分频模式寄存器 00 (PRM00)的第 5 位和第 4 位。
ES110, ES100:　预分频模式寄存器 00 (PRM00)的第 7 位和第 6 位。
CRC001, CRC000:　捕捉/比较控制寄存器 00 (CRC00)的第 1 位和第 0 位。

注意事项　1. TM00 和 CR000 相等时进入的清零&启动模式中，CR000 置非 0000H 值。当此寄存器作为外部事件计数器使用时，不能实现仅对一个脉冲的计数。然而,在自由运行模式和由 TI000 引脚有效沿产生的清零&启动模式中，如果 CR000 置 0000H，TM00 溢出后从 0000H 变为 0001H 时，产生一个中断请求 (INTTM000)。
2. 如果 CR000 的新值小于 16 位定时器计数器 0 (TM00)的值，TM00 继续计数，溢出，然后从 0 开始重新计数。因此，如果 CR000 的新值小于原来的值，在 CR000 的值改变后，定时器必须复位重新启动。
3. 当 16 位定时器/事件计数器 00 停止工作后，CR000 的值是不确定的。
4. 当 CR000 设为比较模式，即使有捕捉触发信号输入，也不响应捕捉操作。
5. 当 P31 引脚作为 TI010 有效沿输入端口使用时，P31 引脚则不能作为定时器输出(TO00)使用。而且，当 P31 用作 TO00 时，也不能作为 TI010 有效沿输入引脚使用。
6. 当 CR000 作为捕捉寄存器使用，如果同时发生寄存器的读操作和输入捕捉触发操作，则捕捉触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕捉触发输入同时发生，捕捉触发为不确定状态。
7. 改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00（17）在定时器操作中改变比较寄存器的注意事项。

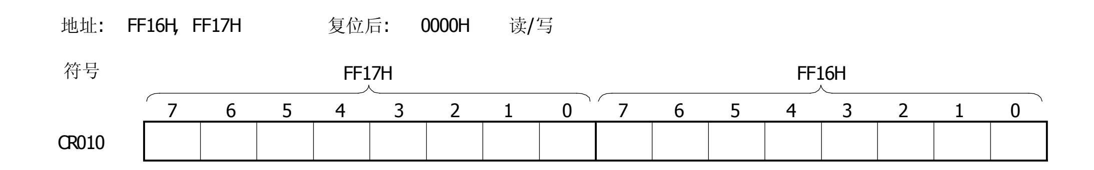
(3) 16 位捕捉/比较寄存器 010 (CR010)

CR010 是可以作为捕捉寄存器和比较寄存器使用的 16 位寄存器。无论是作为比较寄存器使用还是作为捕捉寄存器使用，均由捕捉/比较控制寄存器 00（CR00）的第 0 位（CRC002）来设置。

CR010 由 16 位存储器操作指令设置。

复位后，CR010 值为 0000H。

图 6-4.　16 位定时器捕捉/比较寄存器 010 (CR010)的格式



- 当 CR010 用于比较寄存器

  CR010 中的设置值与 16 位定时器计数器 00（TM00）的计数值不断比较，相等时产生中断请求（INTTM010）。

- 当 CR010 用于捕捉寄存器

  可选择 TI000 引脚的有效沿作为捕捉触发。TI000 的有效沿由预分频模式寄存器 00 设置 (PRM00)（参考 表 6-3）。

表 6-3.　CR010 捕捉触发和 TI000 引脚有效沿 (CRC002 = 1)

CR010 捕捉触发	TI000 引脚有效沿		
		ES010	ES000
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升和下降沿均触发	上升和下降沿均为有效沿	1	1

备注　1. 置 ES010、ES000 = 1, 0 被禁止。

2. ES010, ES000:　预分频模式寄存器 00(PRM00)的第 5 位和第 4 位

   CRC002:　捕捉/比较控制寄存器 00 (CRC00)的第 2 位

注意事项　1. 在自由运行模式和由 TI000 引脚有效沿产生的清零&启动模式中，如果 CR010 置 0000H，TM00 溢出后从 0000H 变为 0001H 时，产生一个中断请求 (INTTM010)。

2. 如果 CR010 新值小于 16 位定时器计数器 0（TM00）的值，TM00 继续计数，溢出，然后从 0 开始重新计数。因此，如果 CR010 新值小于原来的值，则必须在改变赋值后重新启动定时器。

3. 当 16 位定时器/事件计数器 00 停止工作后，CR010 的值是不确定的。

4. 当 CR010 设为比较模式，即使有捕捉触发信号输入，也不响应捕捉操作。

5. 当 CR010 作为捕捉寄存器使用，如果同时发生寄存器的读操作和输入捕捉触发操作，则捕捉触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕捉触发输入同时发生，捕捉触发为不确定状态。

6. 改变 CR010 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00（17）在定时器操作中改变比较寄存器的注意事项。

6.3　16 位定时器/事件计数器 00 的控制寄存器

如下六个类型的寄存器用于控制 16 位定时器/事件计数器 00：

- 16 位定时器模式控制寄存器 00 (TMC00)
- 捕捉/比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频模式寄存器 00 (PRM00)
- 端口模式寄存器 3 (PM3)
- 端口寄存器 3 (P3)

(1) 16 位定时器模式控制寄存器 00 (TMC00)

此寄存器用于设置 16 位定时器的操作模式，16 位定时器计数器 00 (TM00)清零模式，输出时序和检测溢出。

TMC00 由 1 位或 8 位存储器操作指令设置。

复位信号产生后，TMC00 值为 00H。

注意事项　当 TMC002、TMC003 不等于 0，0 时，16 位定时器计数器 00 (TM00)开始工作，而当 TMC002 和 TMC003 等于 0，0 时，停止操作。

图 6-5. 16 位定时器模式控制寄存器 00 (TMC00)的格式

地址: FF60H　复位后值: 00H　读/写

符号	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	操作模式和清零模式选择	TO00 电平反转条件	中断请求发生
0	0	0	操作停止 (TM00 清零)	无反转	无中断产生
0	0	1			
0	1	0	自由运行模式	当 TM00 和 CR000 相等或 TM00 和 CR010 相等时反转	<当作为比较寄存器时> 当 TM00 和 CR000 相等或 TM00 和 CR010 相等时产生中断请求 <当作为捕捉寄存器时> 在 TI000 引脚和 TI010 引脚有效沿产生中断请求
0	1	1		当 TM00 和 CR000 相等；TM00 与 CR010 相等；TI000 引脚有效沿时反转	
1	0	0	在 TI000 引脚有效沿上发生清零 & 启动	-	
1	0	1			
1	1	0	当 TM00 和 CR000 相等时发生清零&启动	当 TM00 和 CR000 相等或 TM00 和 CR010 相等时反转	
1	1	1		当 TM00 和 CR000 相等；TM00 与 CR010 相等；TI000 引脚有效沿时反转	

OVF00	16 位定时器计数器 00 (TM00)的溢出检测
0	无溢出检测
1	检测溢出

注意事项
1. 除了 OVF00 标志位，其他位写入数据时，定时器必须停止操作。
2. 如果定时器停止，即使 TI000/TI010 引脚有信号输入，定时器计数和中断也不响应。
3. 除非当 TI000 引脚有效沿用于计数时钟，在设置定时器 STOP 模式或系统时钟停止之前，必须停止定时器工作，否则，当系统时钟启动时定时器可能产生错误。
4. 当定时器停止工作后才能通过预分频模式寄存器 00(PRM00)的第 4 位和第 5 位设置 TI000 引脚有效沿。
5. 在 TM00 和 CR00 相等时进入的清零&启动模式中，当定时器工作在自由工作模式时，当 CR000 的值为 FFFFH，TM00 的值从 FFFFH 变为 0000H 后，OVF00 标志位置 1。
6. 即便在 TM00 溢出后而下一个计数时钟计数前（TM00 变为 0001H 前）做清除 OVF00 标志操作，OVF00 标志还会被重新置 1， 清除操作无效。
7. 捕捉操作在计数时钟的下降沿执行，中断请求（INTTM0n0）在计数时钟的上升沿产生。

备注　TM00:　16 位定时器计数器 00
CR000:　16 位定时器捕捉/比较寄存器 000
CR010:　16 位定时器捕捉/比较寄存器 010

(2) 捕捉/比较控制寄存器 00 (CRC00)

此寄存器用于控制 16 位捕捉/比较寄存器(CR000, CR010)的操作。

CRC00 由 1 位或 8 位存储器操作指令设置。

复位后，CRC00 值为 00H。

图 6-6. 捕捉/比较控制寄存器 00 (CRC00)的格式

地址： FF62H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 操作模式选择
0	作为比较寄存器
1	作为捕捉寄存器

CRC001	CR000 捕捉触发选择
0	TI010 引脚有效沿时捕捉触发
1	当 TI000 引脚有效沿反向[注]时捕捉触发

CRC000	CR000 操作模式选择
0	作为比较寄存器
1	作为捕捉寄存器

注　如果上升沿和下降沿都被选为引脚 TI000 的有效沿，不执行捕捉操作。

注意事项 1. 设置 CRC00 前，定时器必须停止操作。

2. 当通过 16 位定时器模式控制寄存器 TMC00 选择为 TM00 和 CR000 相等进入的清零&启动模式时，CR000 就不能被指定为捕捉寄存器。
3. 为了确保捕捉操作的稳定，被捕捉的信号宽度要大于 2 个计数时钟(由预分频器模式寄存器 **00 (PRM00)**选择)的长度才能触发捕捉 (参见图 6-17)。

(3) 16 位定时器输出控制寄存器 00 (TOC00)

此寄存器用于控制 16 位定时器/事件计数器输出的操作。它设置定时器的输出 F/F set/reset, 输出反转的 允许/禁止, 16 位定时器/事件计数器 00 的定时器输出允许/禁止, 单脉冲输出操作允许/禁止, 和通过软件设置的单脉冲输出触发。

TOC00 由 1 位或 8 位存储器操作指令设置。

复位后，TOC00 值为 00H。

图 6-7. 16 位定时器输出控制寄存器 00 (TOC00)的格式

地址：FF63H　复位后：00H　R/W

符号	7	〈6〉	〈5〉	4	〈3〉	〈2〉	1	〈0〉
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	由软件控制的单脉冲输出触发
0	无单脉冲输出触发
1	触发单脉冲输出

OSPE00	单脉冲输出控制
0	连续脉冲输出模式
1	单脉冲输出模式[注]

TOC004	当 CR010 与 TM00 相等时定时器输出 F/F 控制位
0	禁止反转操作
1	允许反转操作

LVS00	LVR00	定时器输出 F/F 状态设置
0	0	无改变
0	1	定时器输出 F/F 复位(0)
1	0	定时器输出 F/F 置(1)
1	1	禁止设置

TOC001	当 CR000 与 TM00 相等时定时器输出 F/F 控制位
0	禁止反转操作
1	允许反转操作

TOE00	定时器输出控制
0	禁止输出 (输出置低电平)
1	允许输出

注　单脉冲输出模式只能在自由运行模式和 TI000 引脚作为输入有效沿的清零&开始模式下正常使用。在 TM00 和 CR000 相等时进入的清零&开始模式下，由于溢出不会发生，所以单脉冲输出不可能发生。

注意事项

1. **除了 OSPT00 以外，对其它寄存器设置之前，定时器应停止工作。**
2. **读 LVS00 和 LVR00 时，读出的数据为 0。**
3. **数据被设置后，OSPT00 自动清零，读出的内容是 0 。**
4. **除了单脉冲输出模式以外，不要将 OSPT00 设置为 1。**
5. **当连续设置 OSPT00 位时，需要两个或更多计数时钟周期(由预分频器模式寄存器 00 (PRM00)选择)的写入间隔。**
6. **当 TOE00 为 0 时，设置 TOE00, LVS00, 和 LVR00 要用 8 位存储器操作指令同时设置。 当 TOE00 为 1 时，LVS00 ，LVR00 能够用 1 位存储器操作指令设置。**

(4) 预分频模式寄存器 00 (PRM00)

该寄存器用于设置 16 位定时器计数器 00 (TM00)的计数时钟和 TI000, TI010 引脚输入的有效沿。

PRM00 由 1 位或 8 位存储器操作指令设置。

复位后，PRM00 值为 00H。

图 6-8. 预分频模式寄存器 00 (PRM00)的格式

地址： FF61H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	0
PRM00	ES110	ES100	ES010	ES000	0	0	PRM001	PRM000

ES110	ES100	TI010 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿均为有效沿

ES010	ES000	TI000 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿均为有效沿

PRM001	PRM000	计数时钟选择
0	0	f_{XP} (10 MHz)
0	1	$f_{XP}/2^2$ (2.5 MHz)
1	0	$f_{XP}/2^8$ (39.06 kHz)
1	1	TI000 引脚有效沿[注]

备注 1. f_{XP}: 供给外围设备的时钟振荡频率

2. ( )： f_{XP} = 10 MHz

注 外部时钟要求一个比两个内部计数时钟(f_{XP})周期长的脉冲。

注意事项 1. 停止定时器工作后，向 PRM00 写入数据。

2. 如果 TI000 引脚的有效沿被用于计数时钟，就不能设置清零启动模式和 TI000 引脚有效沿的捕捉。

3. 在下列情况下，注意 TI0n0 引脚有效沿被检测的警告。

〈1〉 系统复位后，如果一个高电平输入 TI0n0 引脚,16 位定时器计数器 00 (TM00)允许操作。

→ 如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。

〈2〉 当 TI0n0 为高电平，TM00 停止工作时，TI0n0 引脚如输入一个低电平则 TM00 开始工作。

→ 如果 TI0n0 的下降沿或上升沿和下降沿作为有效沿，那么 TM00 操作使能后，下降沿会立即检测到。

〈3〉 当 TI0n0 为低电平，TM00 停止工作时，TI0n0 引脚如输入一个高电平则 TM00 开始工作。

→ 如果 TI0n0 的上升沿或上升沿和下降沿作为有效沿，那么 TM00 操作使能后，上升沿会立即检测到。

4. 当 TI000 的有效沿用做计数时钟，和当 TI000 用于捕捉触发器时，用于消除噪声的采样时钟是不同的，对于前者，由 f_{XP} 做计数时钟，对于后者，通过 PRM00 选择的计数时钟。直到采样到有效沿并且连续两次检测到有效电平，捕捉操作才会完成，因此可消除小脉冲宽度的噪声。

5. 当 P31 端口作为有效沿的输入引脚(TI010)使用时，它不能同时作为定时器输出引脚(TO00)使用。当 P31 端口作为定时器输出引脚(TO00)使用时，它不能同时作为有效沿的输入引脚(TI010)使用

备注　n = 0，1

(5) 端口模式寄存器 3 (PM3)

这个寄存器可以位选端口 3 的输入/输出模式。

当 P31/TO00/TI010/INTP2 引脚作为定时器输出时，设置 PM31 和 P31 的输出锁存为 0。

当 P30/TI000/INTP0 和 P31/TO00/TI010/INTP2 引脚作为定时器输入时，设置 PM30 和 PM31 为 1。

此时，P30 和 P31 的输出锁存可能为 0 或 1。

PM3 能用 1 位或 8 位存储器操作指令设置。

复位后设置 PM3 为 FFH。

图 6-9.　端口模式寄存器 3 (PM3)的格式

地址: FF23H　复位后: 0FFH　读/写

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n引脚的输入输出模式选择(n=0或3)
0	输出模式(输出锁存打开)
1	输入模式(输出锁存关闭)

6.4　16位定时器/事件计数器 00 的操作

6.4.1　时间间隔定时器操作

通过设置 16 位定时器模式控制寄存器 00（TMC00）和捕捉/比较控制寄存器 00（CRC00），定时器可作为时间间隔定时器使用，如图 6-11 所示。

设置

基本操作的设置步骤如下：

〈1〉　设置 CRC00 寄存器(见图 6-10)。
〈2〉　设置 CR000 寄存器的值。
〈3〉　通过使用 PRM00 寄存器设置计数时钟。
〈4〉　设置 TMC00 寄存器开始定时器工作（见图 6-10）。

注意事项　当 TM00 工作时，改变 CR000 寄存器的设置可能导致错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17)在定时器操作过程中改变比较寄存器的相关注意事项。

备注　关于怎样允许 INTTM000 中断，请参见**第十三章　中断功能。**

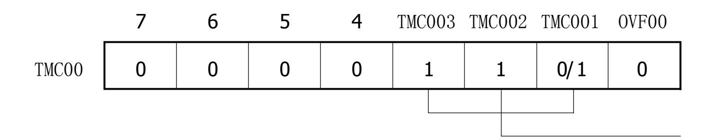
以 16 位定时器捕捉/比较寄存器 000 (CR000) 中预先设定的值为间隔重复产生中断请求。

当 16 位定时器 TM00 的计数值和 CR000 的设定值相等时，TM00 中的值清零，然后继续计数，同时产生中断请求信号(INTTM000)。

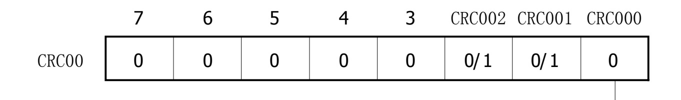
16 位定时器/计数器的计数时钟可以通过预分频模式寄存器 PRM00 的第 0，1 位（PRM000,PRM001）设置。

图 6-10.　设置控制寄存器用于时间间隔定时器操作

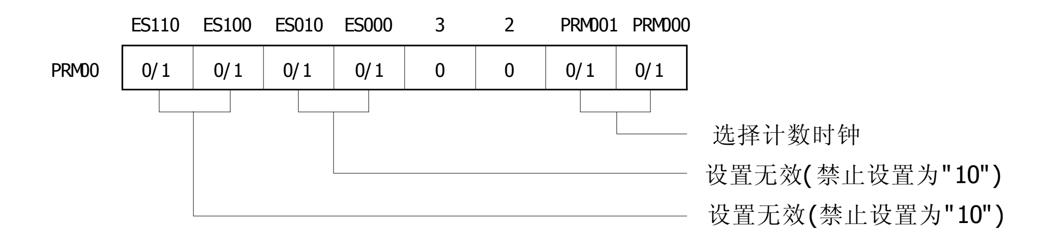
(a)　16 位定时器模式控制寄存器 00 (TMC00)



(b)　捕捉/比较控制寄存器 00 (CRC00)

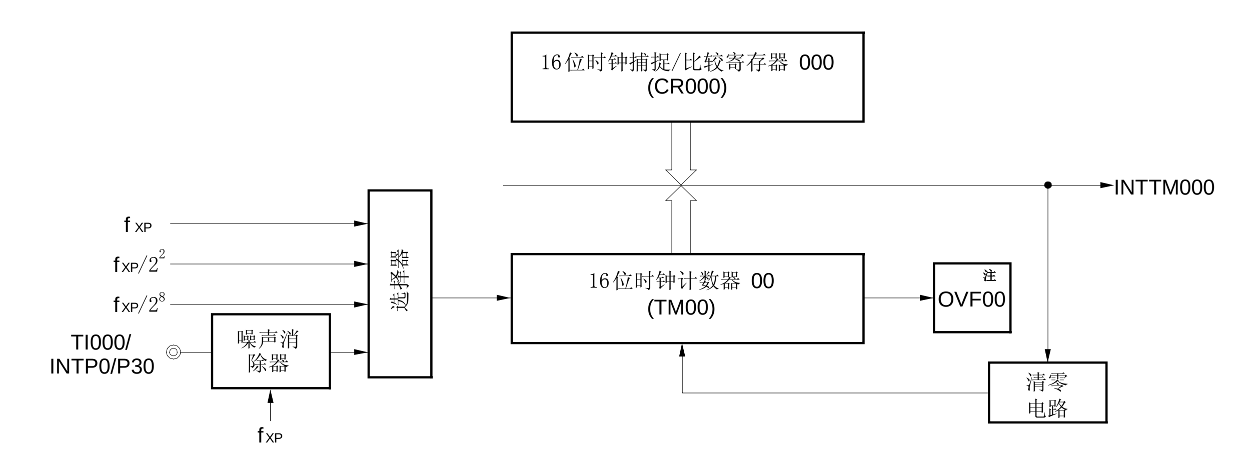


(c)　预分频模式寄存器 00 (PRM00)



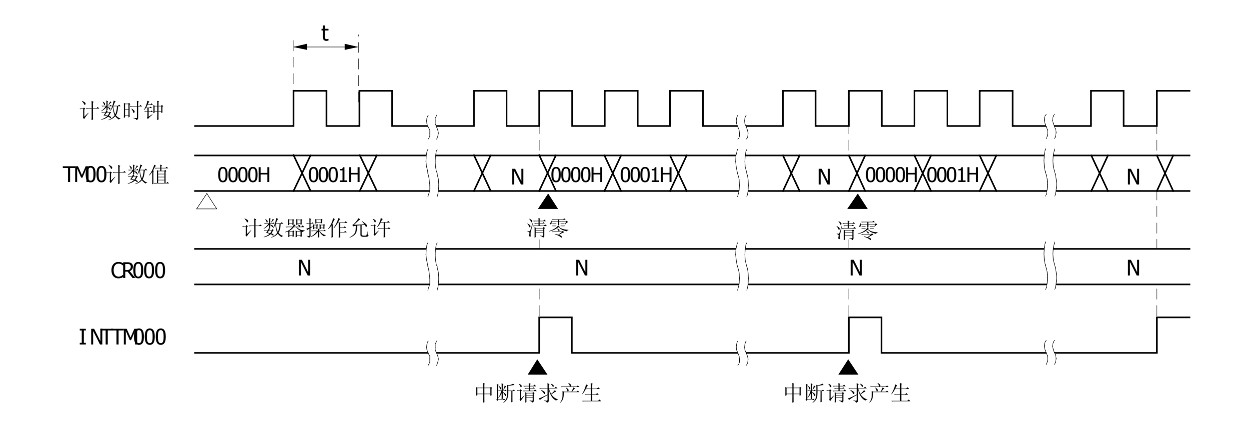
备注　0/1: 设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-11.　间隔定时器配置框图



注　只有当 16 位定时器捕捉/比较寄存器 000（CR000）设置为 FFFFH 时，OVF00 标志位置 1。

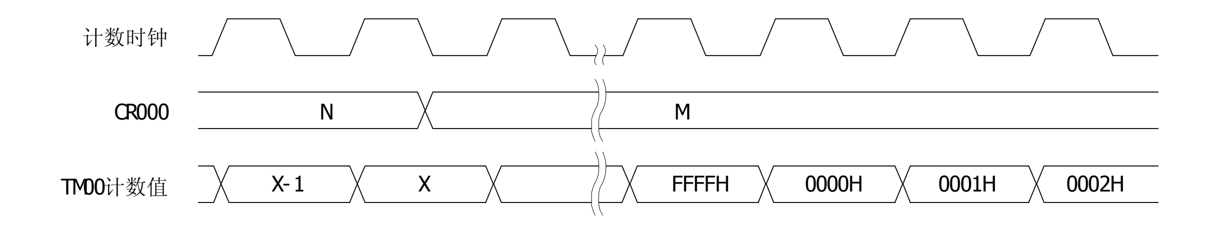
图 6-12.　间隔定时器操作的时序



备注　间隔时间= (N + 1) × t
N = 0001H ～ FFFFH (可设置范围)

当定时器计数过程中比较寄存器被改变时，如果 16 位定时器捕捉/比较寄存器 CR000 改变后的新值小于 16 位定时器 TM00 的值，则 TM00 继续计数，溢出然后从 0 开始计数。因此，如果 CR000 改变后的值(M)小于改变前的值(N),那么在 CR000 改变之后，需要重启定时器。

图 6-13.　在定时器计数操作过程中比较寄存器改变之后的时序 (N → M: N > M)



备注　N > X > M

6.4.2　外部事件计数器操作

设置

基本操作的设置步骤如下：

〈1〉　设置 CRC00 寄存器（见 **图 6-14**）。
〈2〉　通过使用 PRM00 寄存器设置计数时钟。
〈3〉　设置 CR000 寄存器(0000H 不能被设置)。
〈4〉　设置 TMC00 寄存器开始定时器工作(见 **图 6-14**)。

备注　1. 对于 TI000 引脚的设置，请参见 6.3 (5) **端口模式寄存器 3 (PM3)**。
　　　　2. 如何允许 INTTM000 中断,请参见 **第十三章　中断功能**。

外部事件计数器使用 16 位定时器计数器 00（TM00）对 TI000 引脚输入的外部时钟脉冲进行计数。

每当由预分频模式寄存器 00(PRM00)定义的有效沿输入时，TM00 递增。

当 TM00 的计数值与 16 位定时器捕捉/比较寄存器 000 (CR000) 的值相等时，TM00 清零，并且产生中断请求信号 (INTTM000) 。

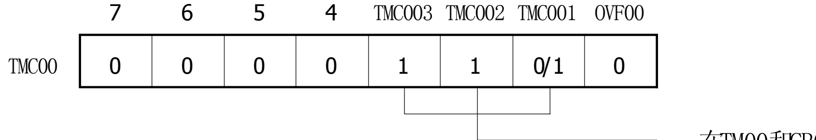
CR000 不能赋值为 0000H (脉冲的计数操作不能完成)。

通过对预分频模式寄存器 PRM00 的第 4，5 位(ES000,ES010)的设置选择上升沿，下降沿或上升沿和下降沿为有效沿。

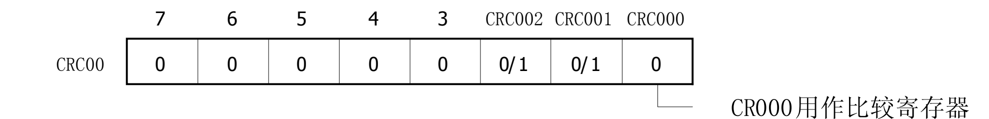
因为仅当使用内部时钟((f_{XP})采样之后检测到两次 TI000 引脚的有效沿时，操作才能进行，所以短脉冲宽度的噪声可以被消除。

图 6-14.　外部事件计数器模式控制寄存器的配置（指定上升沿）

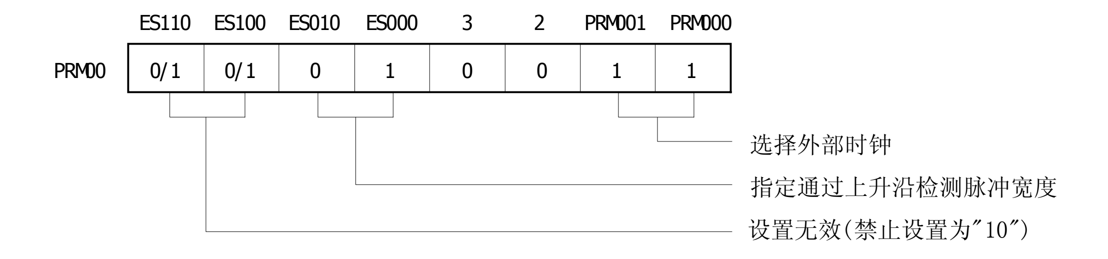
(a)　16 位定时器模式控制寄存器 00（TMC00）



(b)　捕捉/比较控制寄存器 00（CRC00）

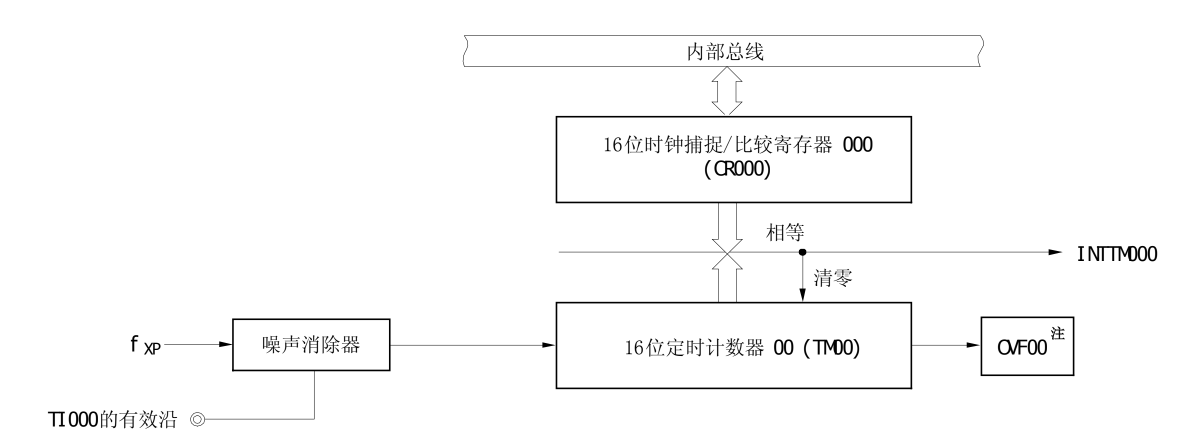


(c)　预分频模式寄存器 00（PRM00）



备注　0/1：设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

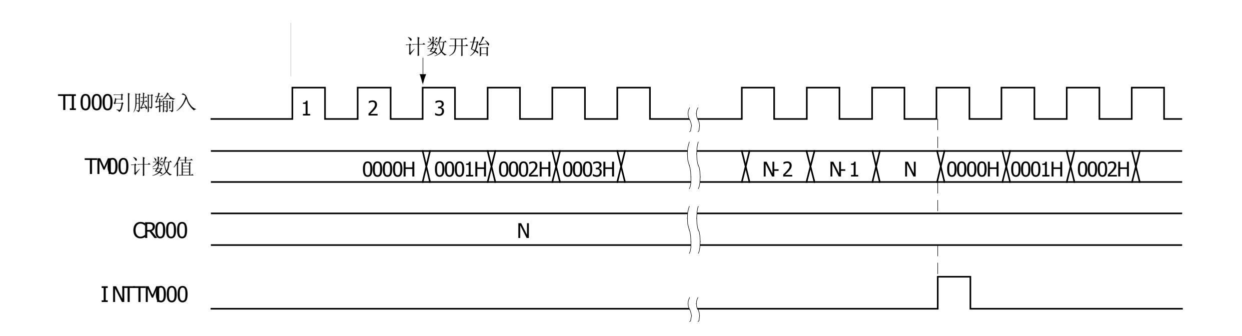
图 6-15.　外部事件计数器配置框图



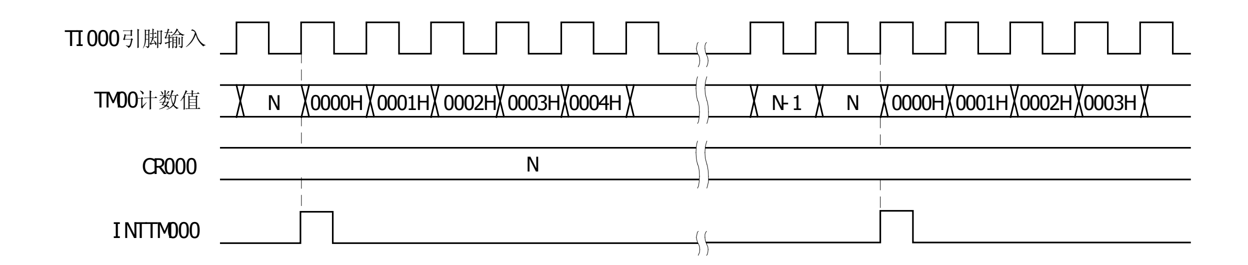
注　只有当 16 位捕捉/比较寄存器 000 中的值为 FFFFH 时，OVF00 标志位置 1。

图 6-16.　外部事件计数器操作时序(指定上升沿)

(1) 操作启动之后立即产生 INTTM000 时序
有效沿被检测到两次之后计数开始



(2) INTTM000 产生两次之后的 INTTM000 产生时序



注意事项　当读取外部事件计数器的计数值时，TM00 即为其值。

6.4.3 脉冲宽度测量操作

通过 16 位定时器计数器 00 (TM00)可测量输入 TI000 和 TI010 引脚的信号的脉冲宽度。

有两种方法可以测量脉冲宽度：一是在自由运行模式下，通过 TM00 测量；二是与输入到 TI000 引脚的信号边沿同步启动定时器测量。

当发生中断时，读捕捉寄存器的有效值，检测溢出标志位，然后计算脉冲宽度。检测后，溢出标志位清零。

直到在由预分频模式寄存器 00 (PRM00)选择的计数时钟周期内采样到信号脉冲宽度，且对 TI000 或 TI010 的有效电平检测过两次之后，才执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-17.　指定上升沿的 CR010 捕捉操作

设置

基本操作的设置步骤如下：

〈1〉　设置 CRC00 寄存器（见 图 6-18，6-21，6-23 和 6-25）。

〈2〉　通过 PRM00 寄存器设置计数时钟。

〈3〉　设置 TMC00 寄存器启动操作（见 图 6-18，6-21，6-23 和 6-25）。

注意事项　　如要使用两个捕捉寄存器，设置 TI000 和 TI010 引脚。

备注　　1. 如需设置 TI000（或 TI010）引脚, 参见 6.3 (5)　端口模式寄存器 3 (PM3)。

2. 如何允许 INTTM000（或 INTTM010）中断，参见 第十三章　中断功能。

(1) 用自由运行计数器和捕捉寄存器测量脉宽

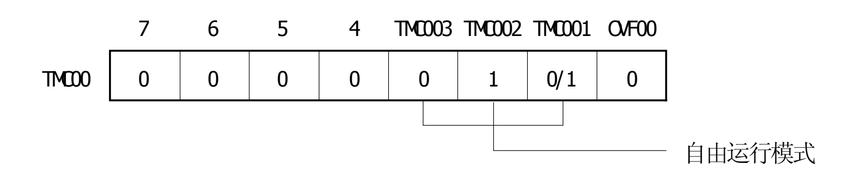
如果16 位定时器计数器00 (TM00)在自由运行模式下工作，且预分频模式寄存器00 (PRM00)规定的脉冲沿输入到TI000 引脚时， TM00 的值被送入16 位定时器捕捉/比较寄存器010(CR010)，并产生外部中断请求信号(INTTM010)。

根据PRM00的第4位和第5位(ES000 和 ES010)指定TI000引脚的上升沿和下降沿同为有效沿。

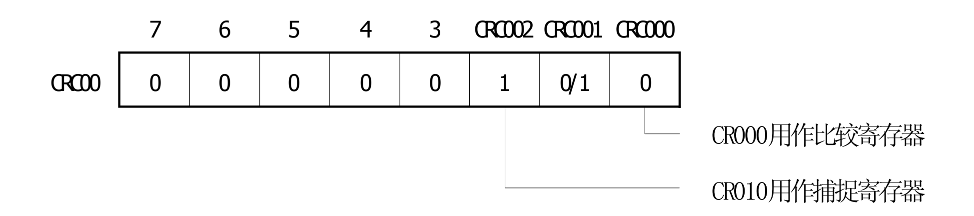
使用 PRM00 选择的计数时钟进行采样，并且仅当 TI000 引脚的有效电平检测到两次才执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-18.　通过自由运行计数器和一个捕捉寄存器测量脉冲宽度的控制寄存器的设置（当使用 TI000 和 CR010 时）

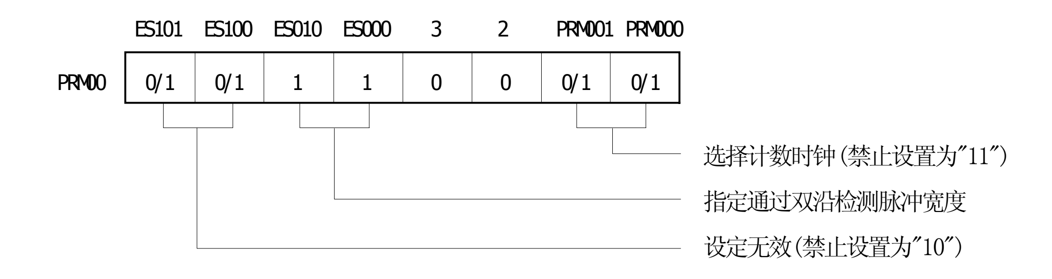
(a)　16 位定时器模式控制寄存器 00 (TMC00)



(b)　捕捉/比较控制寄存器 00 (CRC00)



(c)　预分频模式寄存器 00 (PRM00)



备注　0/1:设置 0 或 1 可允许另外其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-19.　通过自由运行计数器测量脉冲宽度的配置框图

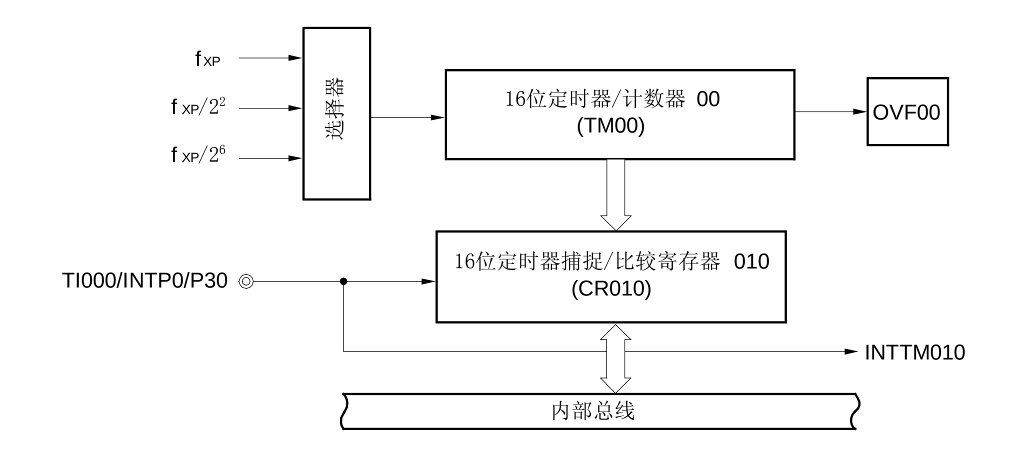
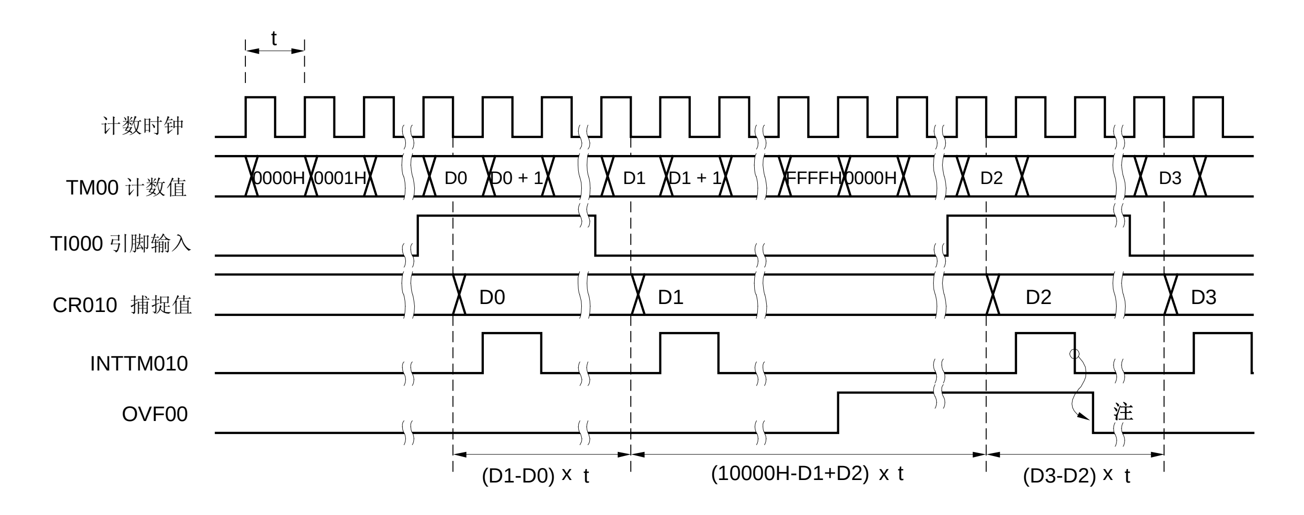


图 6-20.　使用自由运行计数器和一个捕捉寄存器进行脉冲宽度测量的时序（指定上升沿和下降沿）



注　　OVF00 必须通过软件清零。

(2) 使用自由运行计数器测量两个脉冲的脉宽

当 16 位定时器 TM00 工作在自由运行模式下时，定时器 TM00 可以同时测量由 TI000 引脚和 TI010 引脚输入的两个信号的脉冲脉宽。

当有效沿（由 PRM00 的第四位和第五位(ES000 和 ES010)定义）输入到 TI000 引脚时，TM00 的值写入 16 位定时器捕捉/比较寄存器 CR010，然后 INTTM010 中断产生。

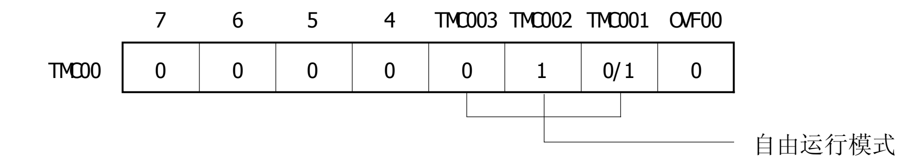
另外，当有效沿（由 PRM00 的第六位和第七位(ES100 和 ES110)定义）输入到 TI010 引脚时，TM00 的值写入 16 位定时器捕捉/比较寄存器 CR000，然后 INTTM000 中断产生。

PRM00 的第 4，5 位（ES000,ES010）和第 6，7 位（ES100,ES110）可以定义 TI000 和 TI010 引脚的上升沿和下降沿都为有效沿。

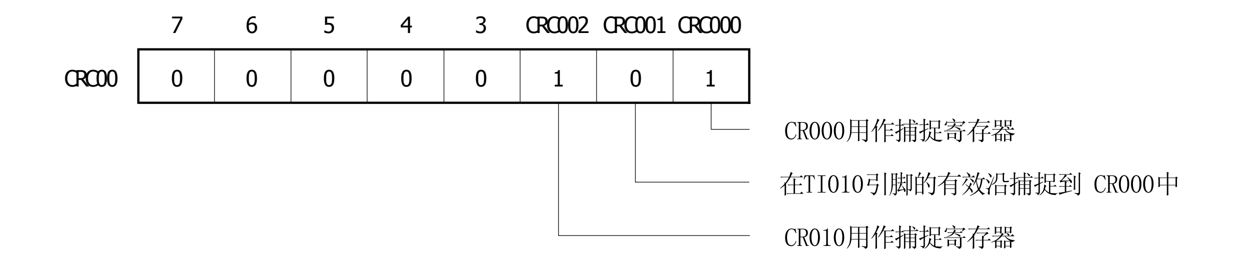
根据预分频模式寄存器 00 (PRM00)选择的计数时钟周期进行采样，并且对 TI000 或 TI010 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-21.　使用自由运行计数器设置控制寄存器用于两个脉冲宽度测量

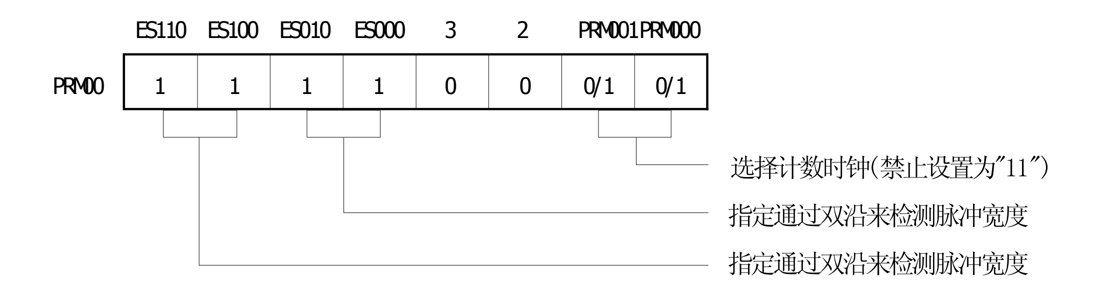
(a)　16 位定时器模式控制寄存器 00 (TMC00)



(b)　捕捉/比较控制寄存器 00 (CRC00)

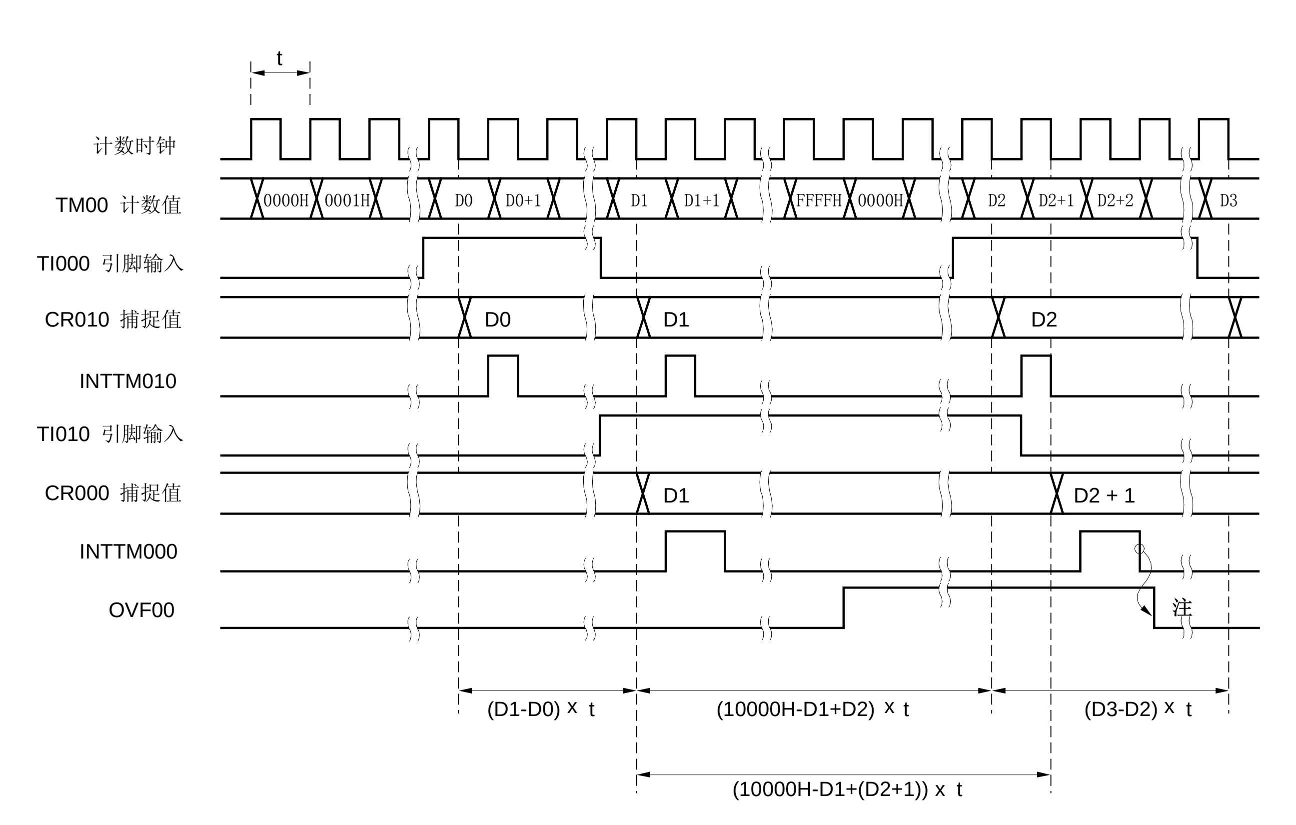


(c)　预分频模式寄存器 00 (PRM00)



备注　0/1:设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-22.　使用自由运行计数器进行脉冲宽度测量的时序（指定上升沿和下降沿）



注　　OVF00 必须通过软件清零。

(3) 使用自由运行计数器和两个捕捉寄存器进行脉冲宽度测量

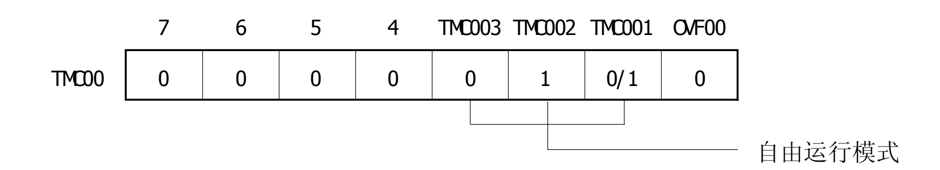
当 16 位定时器 TM00 在自由运行模式下工作时，可以测量由 TI000 引脚输入的信号的脉宽。

当上升沿或者下降沿（由 PRM00 的第四位和第五位(ES000 和 ES010)定义）输入到 TI000 引脚时，TM00 的值写入 16 位定时器捕捉/比较寄存器 CR010，然后 INTTM010 中断产生。

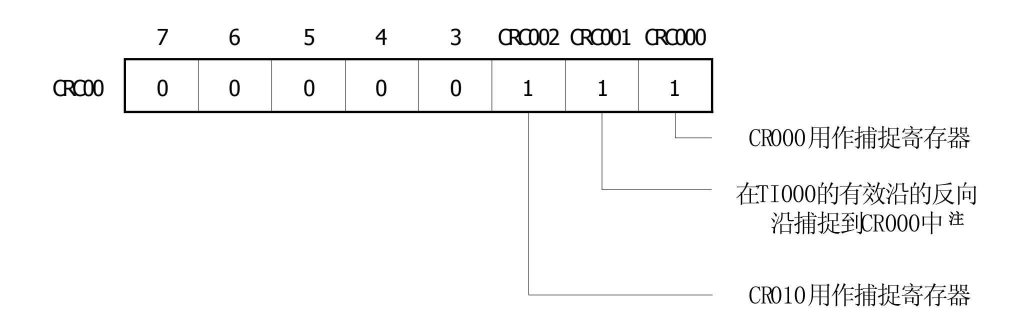
另外，当捕捉 CR010 的有效沿的反向沿输入时， TM00 的值被送入 16 位定时器捕捉/比较寄存器 000(CR000)。根据预分频模式寄存器 00 (PRM00)选择的计数时钟周期进行采样，并且对 TI000 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-23.　设置控制寄存器，使用自由运行计数器和两个捕捉寄存器进行两个脉冲宽度的测量（指定上升沿有效）

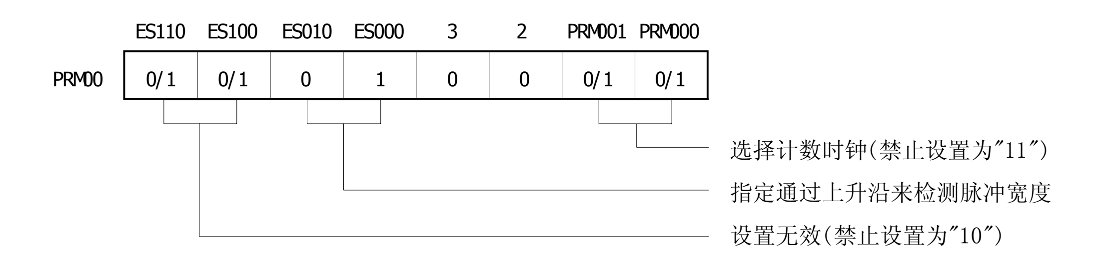
(a)　16 位定时器模式控制寄存器 00 (TMC00)



(b)　捕捉/比较控制寄存器 00 (CRC00)



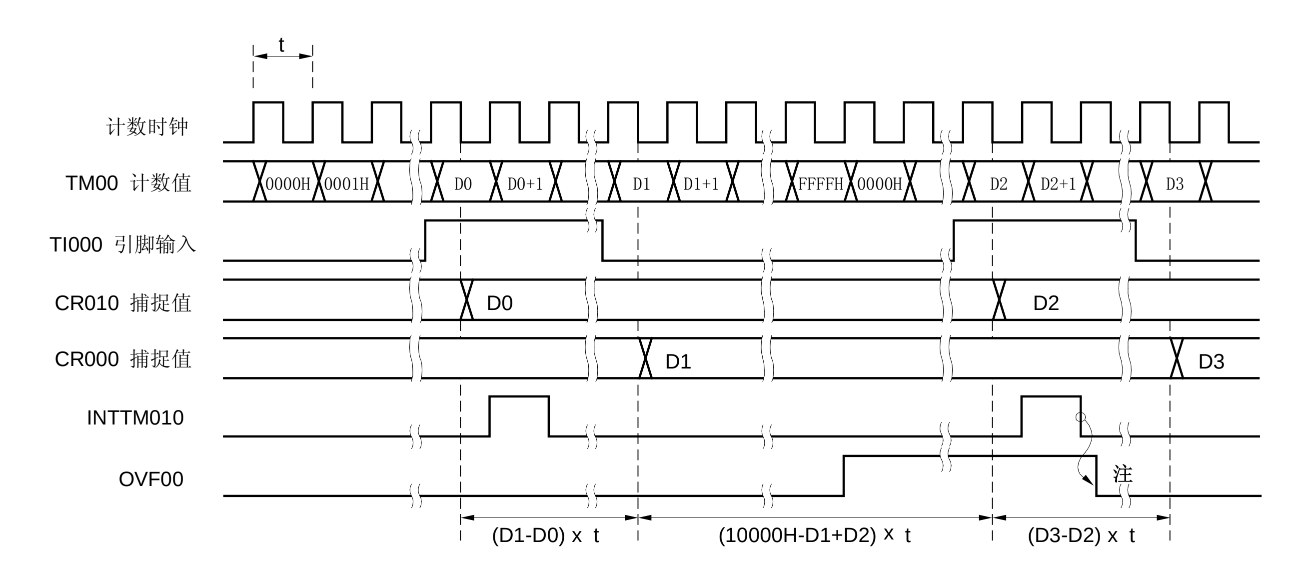
(c)　预分频模式寄存器 00 (PRM00)



注　如果指定 TI000 引脚的上升沿和下降沿同时为有效沿，16 位定时器捕捉/比较寄存器 0000(CR000) 不能完成捕捉操作。当 CRC001 的值为 1，TI010 引脚的有效沿被检测时， TM00 的计数值不能捕捉到 CR000 寄存器中，但是 TI010 引脚的输入可以作为外部中断源，可以产生 INTTM000 中断。

备注　0/1:设置 0 或 1 可允许其他功能与间隔定时器同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-24. 使用自由运行计数器和两个捕捉寄存器进行脉冲宽度测量的时序（指定上升沿有效）



注　　OVF00 必须通过软件清零。

(4) 通过重启进行脉冲宽度测量

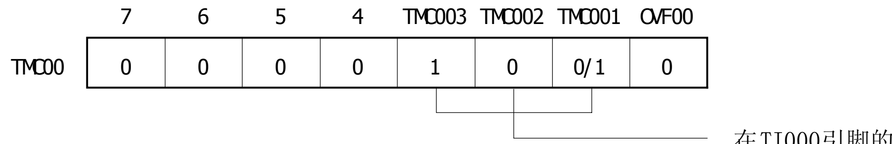
当检测有输入到 TI000 引脚的有效沿时，将 16 位定时器计数器 00(TM00)的计数值送入 16 位定时器捕捉/比较寄存器 010 (CR010)，然后通过对 TM00 清零和重新进行计数操作来测量输入到 TI000 引脚的信号的脉冲宽度。

通过 PRM00 的第 4 位和第 5 位(ES000 和 ES010) 可以指定 TI000 引脚的上升沿和下降沿同为有效沿。

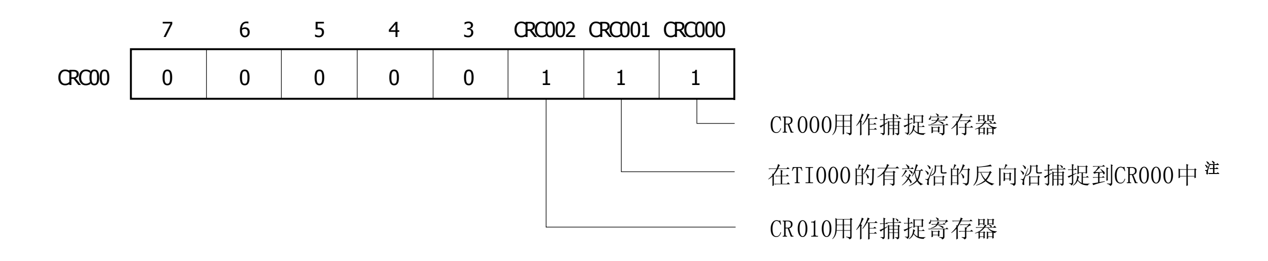
根据预分频模式寄存器 00 (PRM00)选择的计数时钟周期进行采样，并且对 TI000 的有效电平检测过两次之后才可执行捕捉操作，因此可以消除短脉冲宽度的噪音。

图 6-25.　通过重启定时器设置脉冲宽度测量的控制寄存器（指定上升沿有效）

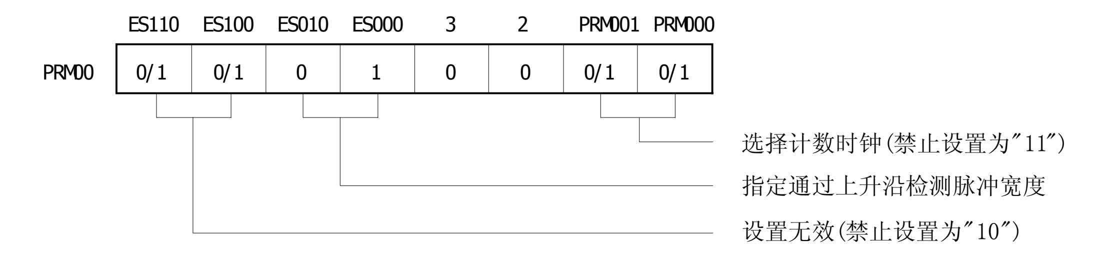
(a)　16 位定时器模式控制寄存器 00 (TMC00)



(b)　捕捉/比较控制寄存器 00 (CRC00)

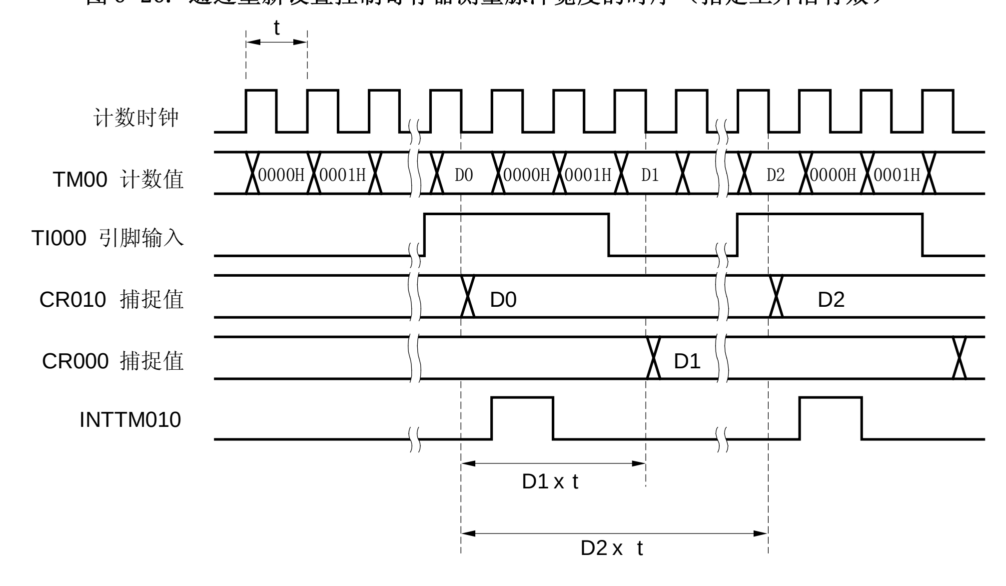


(c)　预分频模式寄存器 00 (PRM00)



注　如果指定 TI000 引脚的上升沿和下降沿同为有效沿，16 位定时器捕捉/比较寄存器 0000(CR000) 不能完成捕捉操作。

图 6-26. 通过重新设置控制寄存器测量脉冲宽度的时序（指定上升沿有效）



6.4.4 方波输出操作

设置

基本操作的设置步骤如下：

<1> 通过 PRM00 寄存器设置计数时钟。

<2> 设置 CRC00 寄存器（见 图 6-27）。

<3> 设置 TOC00 寄存器（见 图 6-27）。

<4> 设置 CR000 寄存器（不能设置成 0000H）。

<5> 设置 TMC00 寄存器启动操作（见 图 6-27）。

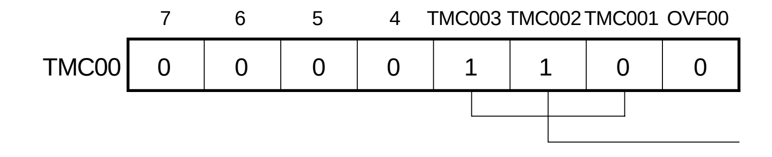
注意事项 如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 **00 (17)**在定时器操作过程中改变比较寄存器的相关注意事项。

备注 1. 如需设置 TO00 管脚，请参见 6.3 (5) **端口模式寄存器 3 (PM3)**。
2. 如何允许 INTTM000 中断，请参见 **第十三章 中断功能**。

根据 16 位定时器捕捉/比较寄存器 000(CR000)预置计数值所确定的时间间隔，可输出任选频率方波。
通过将16 位定时器输出控制寄存器00 (TOC00)的第0 位(TOE00)和第1 位(TOC001)设置为1，可使TO00引脚输出状态以 CR00n＋1 的预置值为间隔翻转。这样可输出任选频率的方波。

图 6-27. 方波输出模式下控制寄存器的设置 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉比较控制寄存器 00 (CRC00)

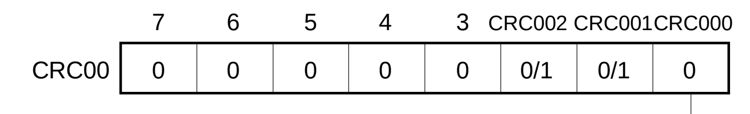
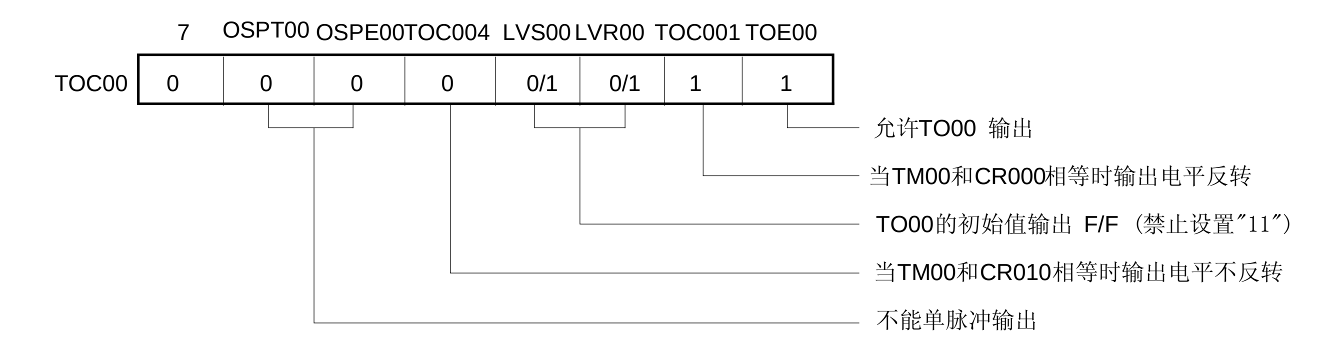
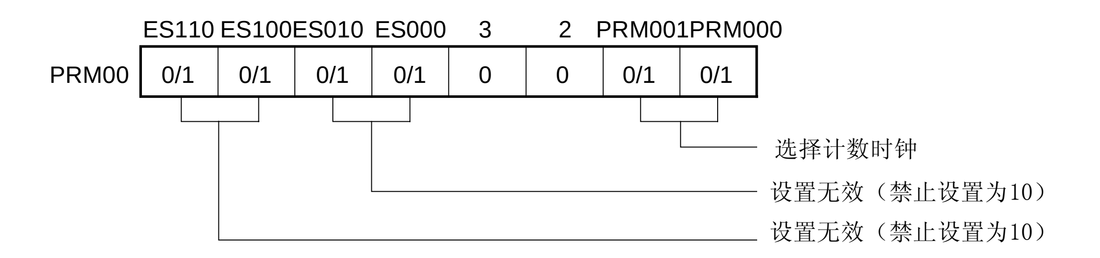


图 6-27.　方波输出模式下控制寄存器的设置 (2/2)

(c)　16 位定时器输出控制寄存器 00 (TOC00)

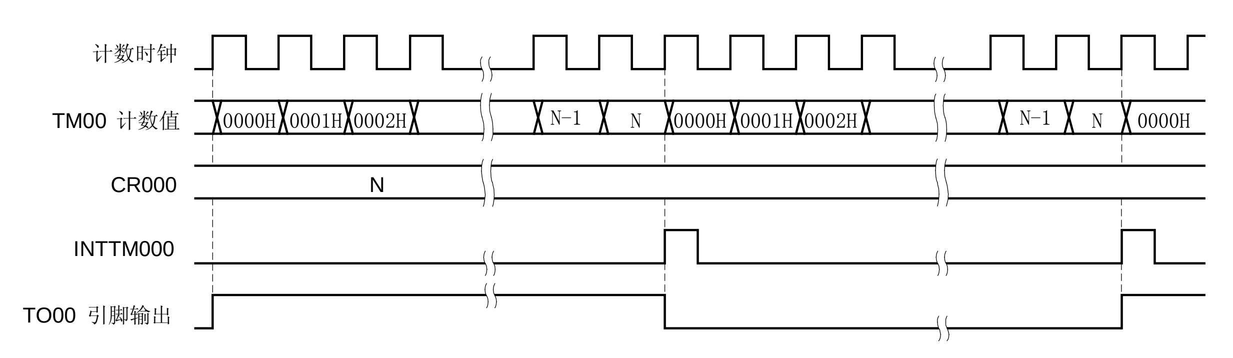


(d)　预分频模式寄存器 00 (PRM00)



备注　0/1:设置 0 或 1 可允许其他功能与方波输出同时使用。需要了解详细信息，参见各自控制寄存器的描述。

图 6-28.　方波输出操作时序



6.4.5 PPG 输出操作

通过设置 16 位定时器模式控制寄存器（TMC00）和捕捉/比较控制寄存器（CRC00）可以使用可编程脉冲发生器 PPG 输出，详见图 6-29。

设置

基本操作的设置步骤如下：

<1> 设置 CRC00 寄存器（见 图 6-29）。
<2> 设置 CR000 寄存器的值，作为时钟周期。
<3> 设置 CR010 寄存器为任意值，作为占空系数。
<4> 设置 TOC00 寄存器（见 图 6-29）。
<5> 通过 PRM00 寄存器设置计数时钟。
<6> 设置 TMC00 寄存器启动操作（见 图 6-29）。

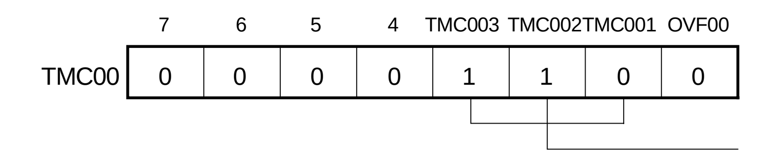
注意事项 **如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17)在定时器操作过程中改变比较寄存器的相关注意事项。**

备注 1. 如需设置 TO00 引脚，请参见 6.3 (5) **端口模式寄存器 3 (PM3)**。
2. 如何允许 INTTM000 中断，请参见 **第十三章 中断功能**。
3. n = 0 或 1

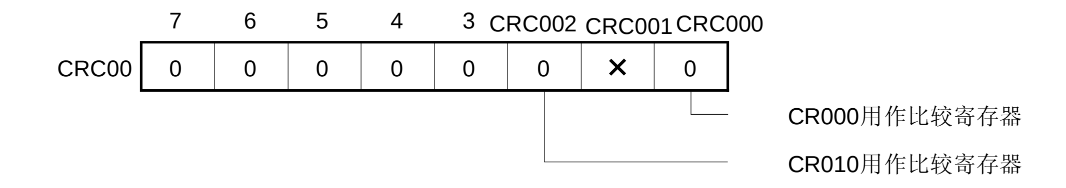
在 PPG 输出操作中，从 TO00 引脚输出矩形波，脉冲宽度和周期分别与 16 位定时器捕捉/比较寄存器 010 (CR010) 和 16 位定时器捕捉/比较寄存器 000 (CR000)中的预置计数值一致。

图 6-29. PPG 输出操作的控制寄存器的设置

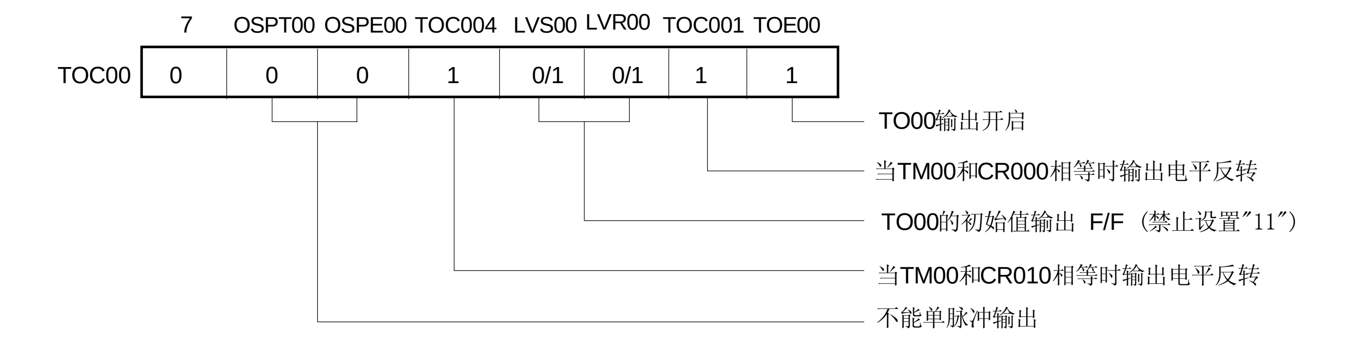
(a)　16 位定时器模式控制寄存器 00 (TMC00)



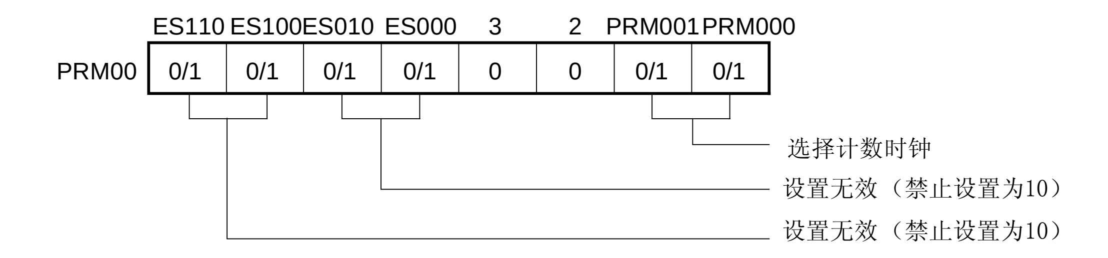
(b)　捕捉/比较控制寄存器 00 (CRC00)



(c)　16 位定时器输出控制寄存器 00 (TOC00)



(d)　预分频模式寄存器 00 (PRM00)



注意事项
1. 以下为 CR000 和 CR010 的取值范围：
   0000H < CR010 < CR000 ≤ FFFFH
2. 由 PPG 输出(CR00n 设置值+ 1)产生的脉冲周期的占空比为 (CR01n 设置值 + 1)/(CR00n 设置值 + 1)。

备注　×: 不必考虑

图 6-30.　PPG 输出的配置框图

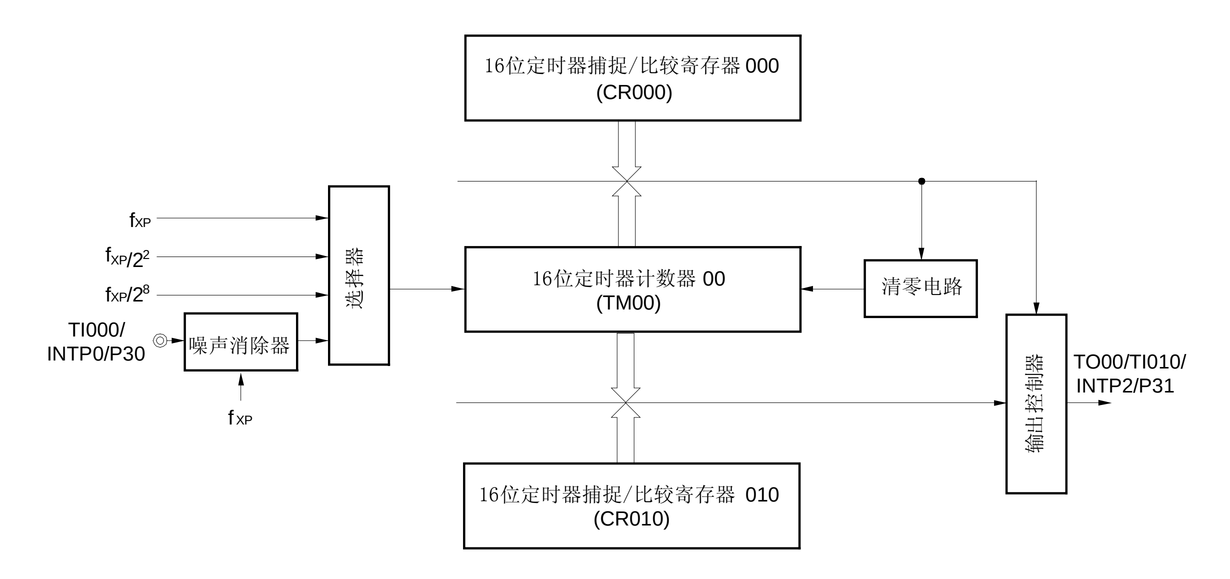
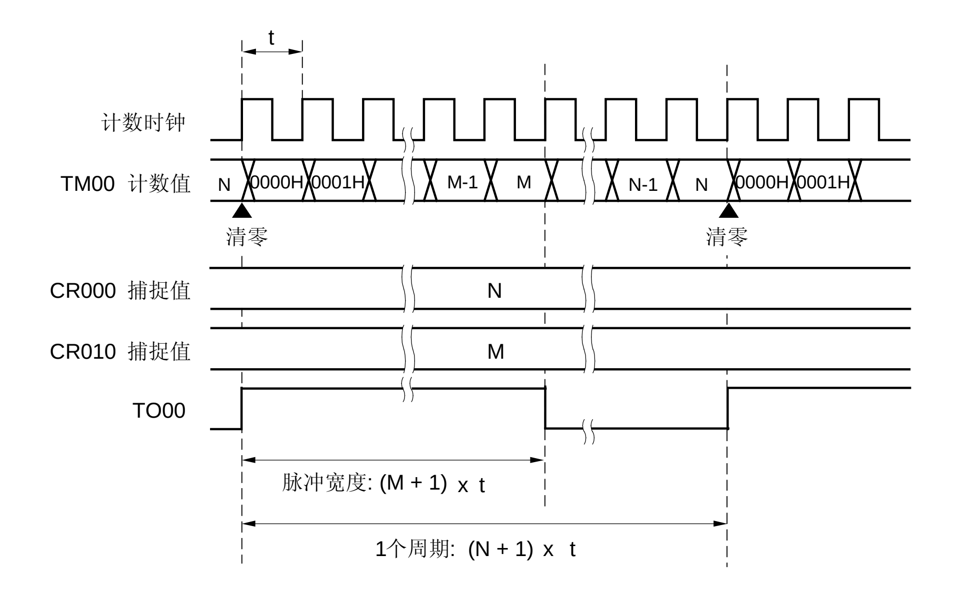


图 6-31.　PPG 输出操作的时序



备注　　0000H < M < N ≤ FFFFH

6.4.6 单脉冲输出操作

16 位定时器/事件计数器 00 可与软件触发或外部触发(TI000 引脚输入)同步输出单脉冲。

设置

基本操作的设置步骤如下：

〈1〉 通过 PRM00 寄存器设置计数时钟。
〈2〉 设置 CRC00 寄存器(设置值参考图 6-32 和 6-34)。
〈3〉 设置 TOC00 寄存器(设置值参考图 6-32 和 6-34)。
〈4〉 设置 CR000 和 CR010 寄存器的值（不可设置成 0000H）。
〈5〉 设置 TMC00 寄存器启动操作(设置值参考图 6-32 和 6-34)。

备注 1. 如需设置 TO00 引脚, 参见 6.3 (5) **端口模式寄存器 3 (PM3)**.
2. 如何允许 INTTM000 中断（根据需要可选 INTTM010），请参见 **第十三章 中断功能。**

(1) 通过软件触发进行单脉冲输出

通过设置 16 位定时器模式控制寄存器 00（TMC00），捕捉/比较控制寄存器 00（CRC00），16 位定时器输出控制寄存器 00（TOC00），通过软件将 TOC00 寄存器的第 6 位设置成 1，可以输出单脉冲信号，如下图 6-32 所示。

将 OSPT00 设置为 1，可对 16 位定时器/事件计数器 00 清零，并启动。并且计数值与预先设置到 16 位定时器捕捉/比较寄存器 010(CR010)的值(N)相等后，输出有效。而在计数值与预先设置到 16 位定时器捕捉/比较寄存器 000(CR000)的值(M) 相等后，输出无效。[注]

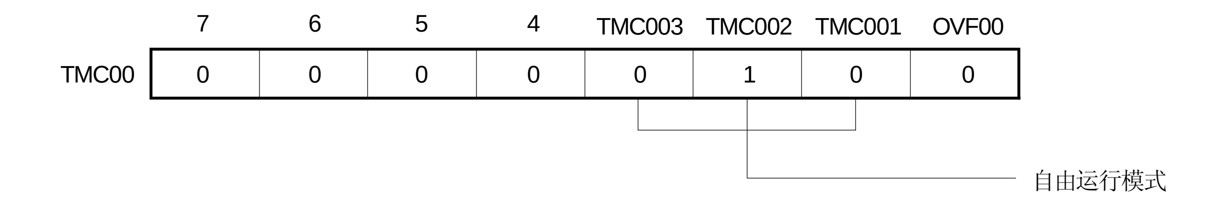
单脉冲输出后，寄存器 TM00 仍然在工作。必须将 TMC00 寄存器的 TMC003 和 TMC002 清零才能中止 TM00 寄存器操作。

注： 这里所描述的是 N < M 的情况。 若 N > M ，计数值与 CR000 相等后输出有效，而与 CR010 相等后输出无效。不能将 N 等于 M。

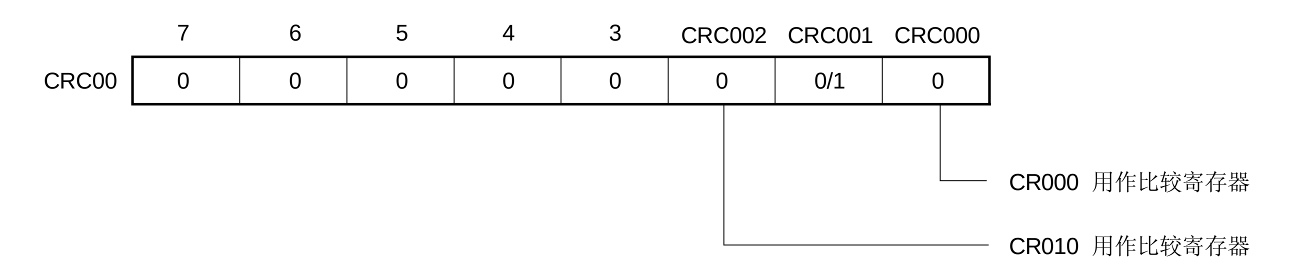
注意事项 1．当单脉冲输出时，不要设置 OSPT00 位。在当前单脉冲输出完成后才可再次输出单脉冲。
2．使用软件触发进行 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 TI000 及其复用端口引脚的电平。由于在这种情况下外部触发也有效，所以可使用 TI000 或其复用端口引脚对定时器清零&启动。这样会输出非预期的时序。

图 6-32.　软件触发的单脉冲输出的控制寄存器的设置

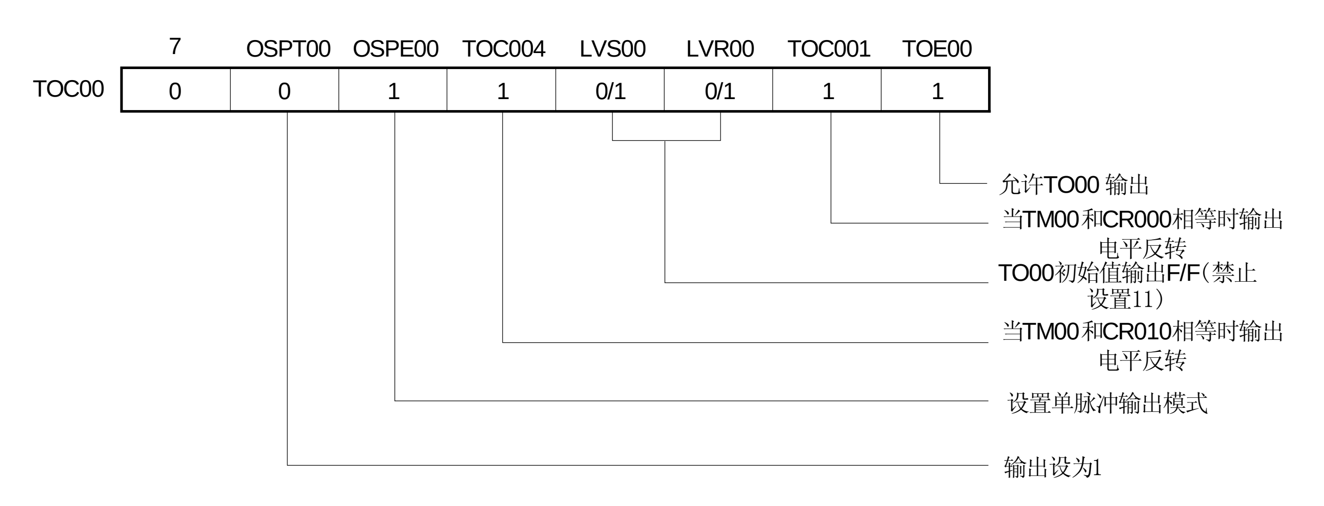
(a)　16 位定时器模式控制寄存器 00（TMC00）



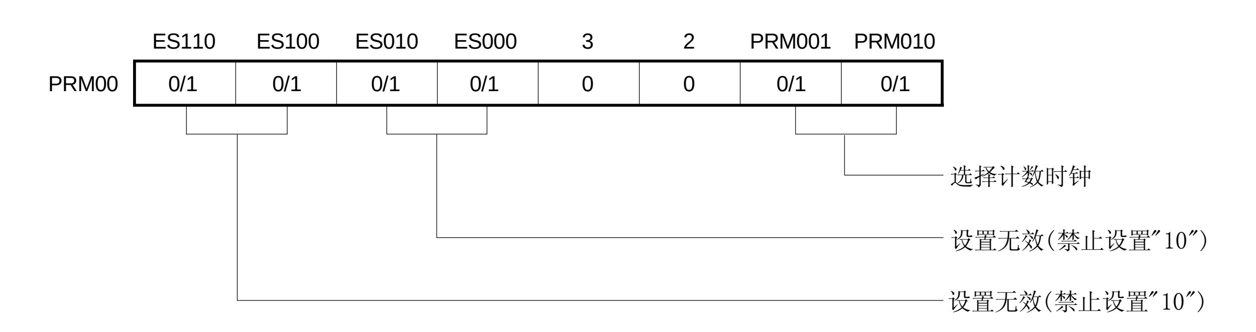
(b)　捕捉/比较控制寄存器 00（CRC00）



(c)　16 位定时器输出控制寄存器 00（TOC00）

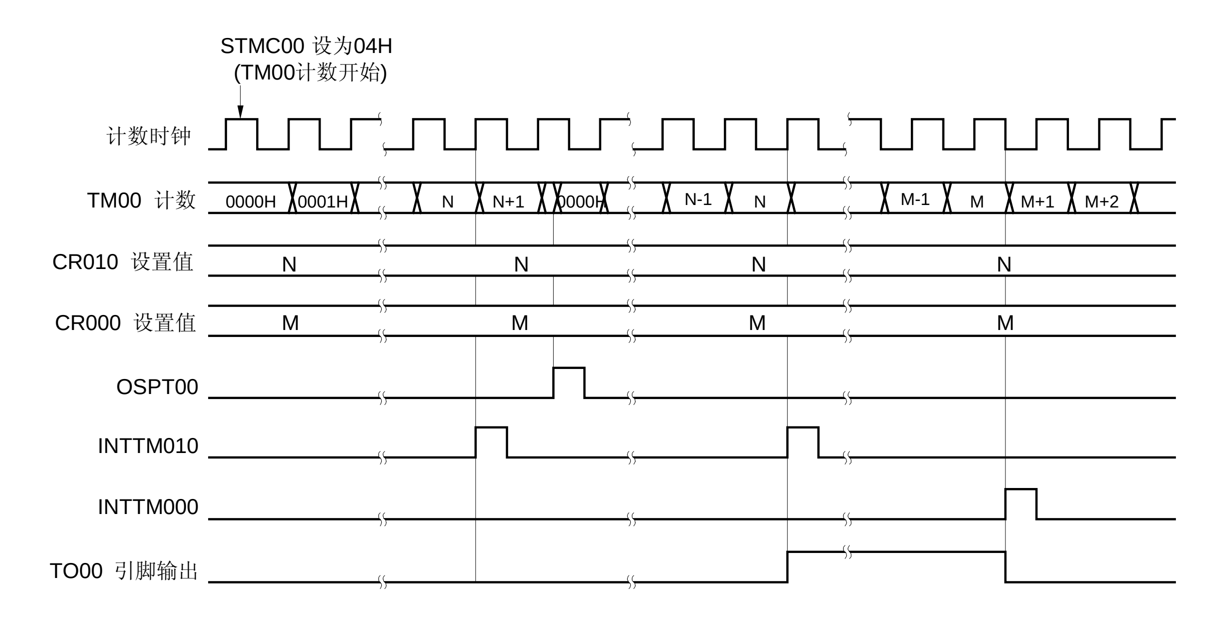


(d)　预分频模式寄存器 00（PRM00）



注意事项　**CR000** 和 **CR010** 寄存器不能置 **0000H**。

图 6-33. 软件触发的单脉冲输出操作的时序



注意事项　**一旦为 TMC003 和 TMC002 位赋值（00 除外，操作停止模式），16 位定时器计数器 00 开始计数。**

备注　N < M

(2)　外部触发的单脉冲输出

通过设置 16 位定时器模式控制寄存器 00（TMC00），捕捉/比较控制寄存器 00（CRC00），16 位定时器输出控制寄存器 00（TOC00）和使用 TI000 引脚的有效沿作为外部触发，可以从 TO00 引脚输出单脉冲信号，如下图 6-34 所示。

预分频模式寄存器 00（PRM00）寄存器的第 4，5 位(ES000 和 ES010)可以设定 TI000 引脚的有效沿。有效沿可以为上升沿，下降沿或皆有上升沿和下降沿三种。

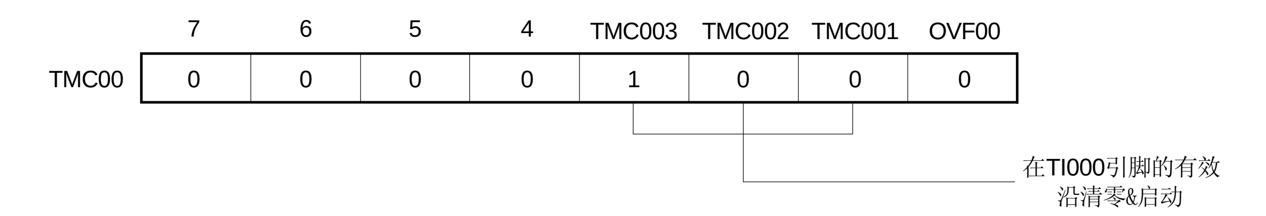
当检测到 TI000 的有效沿时，将 16 位定时器/事件计数器清零并开始计数。并且计数值与预先设置到 16 位定时器捕捉/比较寄存器 010(CR010)的值(N)相等后，输出有效。而在计数值与预先设置到 16 位定时器捕捉/比较寄存器 000(CR000)注的值(M) 相等后，输出无效。

注： 这里所描述的是 N < M 的情况。 若 N > M ，计数值与 CR000 相等后输出输出有效，而与 CR010 相等后输出无效。不能将 N 等于 M。

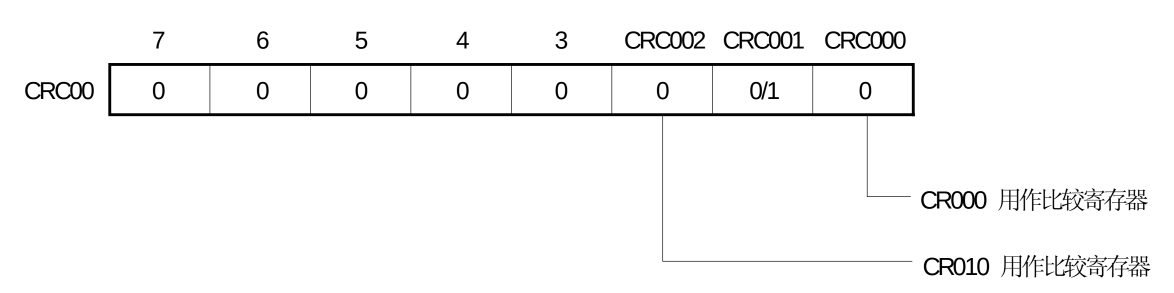
注意事项　当单脉冲正在输出时不要再次再输入外部触发。在当前单脉冲输出完成后才可再次输出单脉冲。

图 6-34. 外部触发的单脉冲输出的控制寄存器设置
(指定上升沿有效)

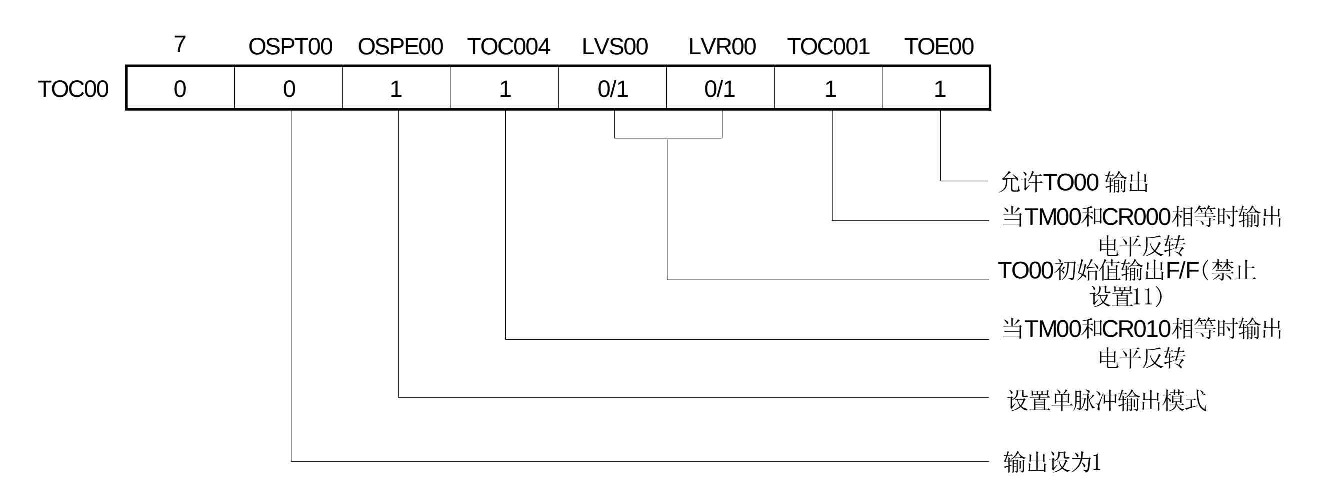
(a) 16 位定时器模式控制寄存器 00 (TMC00)



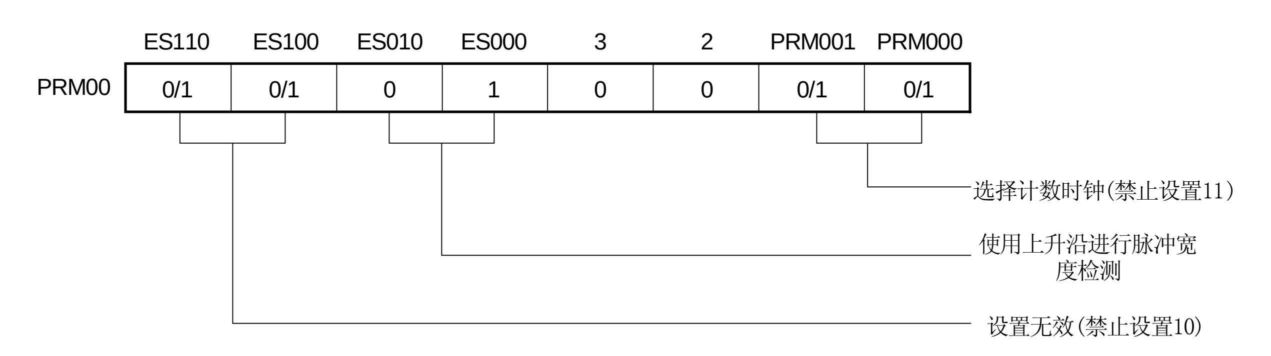
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频模式寄存器 00 (PRM00)



注意事项 **CR000** 和 **CR010** 寄存器不能置 **0000H** 。

图 6-35.　外部触发单脉冲输出的时序（指定上升沿有效）

注意事项　　一旦为 TMC003 和 TMC002 位赋值（00 除外，操作停止模式），16 位定时器计数器 00 开始计数。

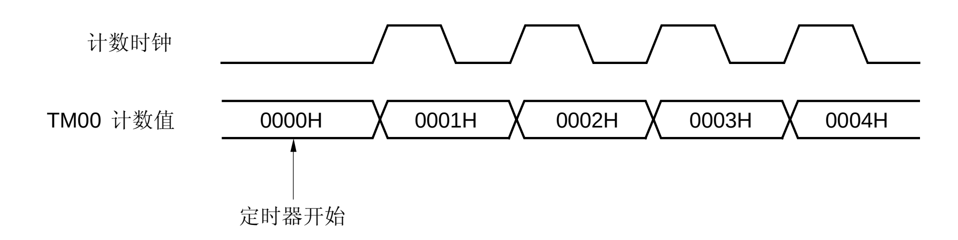
备注　　N < M

6.5　有关 16 位定时器/事件计数器 00 的注意事项

(1) 定时器启动误差

定时器启动后在信号匹配产生之前可能发生将近一个时钟的误差，这是由于 16 位定时器计数器 00 (TM00) 的启动与计数时钟不同步。

图 6-36.　16 位定时器计数器 00 (TM00)的启动时序



(2) 16 位定时器计数器 00 (TM00) 操作

〈1〉 当 TMC002 和 TMC003 位的值不为 0，0（操作停止模式）时，16 位定时器 TM00 开始工作。相反，当 TMC002 和 TMC003 位的值设置为 0，0 时，16 位定时器 TM00 停止工作。

〈2〉 即使 TM00 的值可读，这个值也不能被 16 位定时器捕捉/比较寄存器 010（CR010）捕捉。

〈3〉 当 TM00 中的值被读取时，计数时钟停止工作。

〈4〉 不管 CPU 处于什么工作模式，当定时器是停止的，输入 TI000/TI010 引脚的信号不能被响应。

(3) 16 位定时器捕捉/比较寄存器 000, 010 (CR000, CR010) 的设置

〈1〉 在 TM00 和 CR000 相等时进入清零&启动模式中，通过设置 16 位捕捉/比较寄存器 000（CR000）为非 0000H 的值。也就是说当此寄存器作为外部事件计数器时，不能执行 1 个脉冲计数操作。

〈2〉 TM00 和 CR000 相等时进入清零&启动模式，CR000 寄存器不能作为一个捕捉寄存器使用。

〈3〉 在自由运行模式和清零&启动工作模式下 TI000 引脚输入有效沿作为输入时，如果 CR0n0 寄存器设置为 0000H，当 CR0n0 从 0000H 变化到 0001H 时，INTTM0n0 中断产生。

〈4〉 如果 CR0n0 的新值比 TM00 的值小时，TM00 继续计数，溢出，然后从 0 开始计数。因此，如果 CR0n0 的新值比老值小时，定时器必须在改变了 CR0n0 的值以后重新启动。

(4) **捕捉寄存器的数据保持**

当 16 位定时/计数器 TM00 停止工作后，16 位定时器捕捉/比较寄存器 CR0n0 的值不确定。

备注　　n = 0, 1

(5) **16 位定时器模式控制寄存器 00（TMC00）的设置**

在向 16 位定时器模式控制寄存器 TMC00 写入数据前，定时器必须停止工作，OVF00 标志位除外。

(6) **捕捉/比较控制寄存器 00 (CRC00)的设置**

设置 CRC00 时，定时器操作必须停止。

(7) **16 位定时器输出控制寄存器 00 (TOC00)的设置**

〈1〉　设置定时器前，必须停止工作，OSPT00 除外。

〈2〉　当读取 LVS00 和 LVR00 时，读出值为 0。

〈3〉　数据设置之后，OSPT00 自动清零，读该位时，读出值为 0。

〈4〉　除单脉冲输出模式外，OSPT00 不可设置为 1。

〈5〉　当连续设置 OSPT00 为 1 时，需要用预分频模式寄存器 00（PRM00）选择两个或更多计数时钟周期的写入间隔。

(8) **预分频模式寄存器 00（PRM00）的设置**

定时器停止工作后，向 PRM00 写入数据。

(9) **有效沿设置**

定时器停止工作后，通过设置预分频模式寄存器 00（PRM00）的第 4，5 位(ES000 和 ES010)可设置 TI000 引脚的输入有效沿。

(10) **单脉冲输出**

一般只有在自由运行模式或清零&启动模式下，TI000 引脚有效沿输入才能有单脉冲输出。因为在清零&启动模式下， TM00 和 CR000 相等时不发生溢出，所以不可能输出单脉冲。

(11) **软件设置的单脉冲输出**

〈1〉　在单脉冲正在输出情况下，不要再次置 OSPT00 位为 1。等正在进行的单脉冲输出完毕后再次输出单脉冲。

〈2〉　当用软件触发 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 T1000 引脚或其复用端口引脚的电平。因为即使在这种情况下外部触发也是有效的，即使是 T1000 引脚或其复用端口引脚的电平也能使定时器清零并重新开始。在非预期的时间内产生脉冲输出。

〈3〉　16 位定时器捕捉/比较寄存器 000 和 010（CR000 和 CR010）不能设置为 0000H。

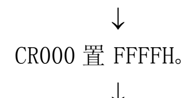
(12) 外部触发的单脉冲输出

〈1〉 在单脉冲输出进行时不要再次输入外部触发。等正在进行的单脉冲输出完毕后再输出单脉冲。

〈2〉 16 位定时器捕捉/比较寄存器 000 和 010（CR000 和 CR010）不能设置为 0000H。

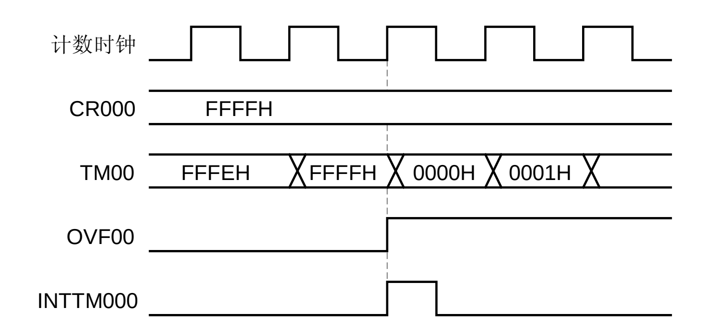
(13) OVF00 标志位操作

〈1〉 OVF00 标志位在下列情况下设置为 1。
在选择清零&启动模式（由 TM00 和 CR000 相等或者在 T1000 的有效沿时进入），或选择自由运行模式时。



↓
CR000 置 FFFFH。
↓
当 TM00 从 FFFFH 计数到 0000H 时。

图 6-37.　OVF00 标志位的操作时序

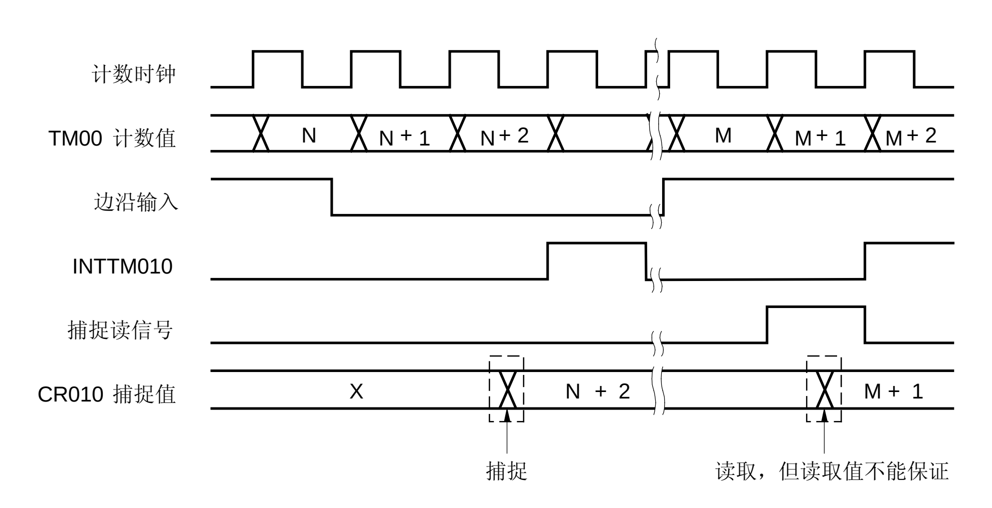


〈2〉 即便在 TM00 溢出后而下一个计数时钟计数前（TM00 变为 0001H 前）做清除 OVF00 标志操作， OVF00 标志还会被重新置 1， 清除操作无效。

(14) 冲突操作

CR000/CR010 被用作为捕捉寄存器时，如果寄存器读周期和捕捉触发输入相冲突，那么捕捉触发输入有优先权，被读数据不确定。同样，当计数终止输入和捕捉触发输入冲突时，被捕捉数据不确定。

图 6-38.　捕捉寄存器数据保留时序



(15) 捕捉操作

〈1〉 如果指定 TI000 引脚作为计数时钟的有效沿,就不能通过捕捉寄存器设置 TI000 引脚执行捕捉操作。

〈2〉 如果 TI000 引脚的上升沿和下降沿同时作为输入有效沿，则捕捉操作无效。

〈3〉 当 CRC001 值为 1，如果 TI010 引脚有效沿被检测， TM00 的计数值不能被捕捉到 CR000 寄存器中，但是 TI010 引脚的输入可以用作外部中断源，因为可由外部有效信号输入产生 INTTM000 中断。

〈4〉 为保证捕捉操作的可靠性，捕捉触发需要用预分频模式寄存器 00（PRM00）来选择一个大于两个计数时钟周期的脉冲。

〈5〉 在计数时钟的下降沿处完成捕捉操作。然而，在下个计数时钟的上升沿会出现一个中断请求输入(INTTM0n0)。

〈6〉 要使用两个捕捉寄存器时，设置 TI000 和 TI010 引脚。

(16) 比较操作

当 CR0n0 在比较模式下，即使有捕捉触发输入，捕捉操作也可能不能执行。

备注　　n = 0, 1

(17) 在定时器工作时改变比较寄存器的值

〈1〉 16 位定时器捕捉/比较寄存器 0n0（CR0n0）作为比较寄存器使用，在定时器计数期间，在 16 位定时器计数器 00（TM00）和 16 位定时器捕捉/比较寄存器 0n0（CR0n0）的值相等的时序附近修改 CR0n0 的值时，改变 CR0n0 时序和相等时序可能会冲突，这种情况下，操作不能得到保证。在定时器计数时改变 CR0n0 的值时，INTTM000 中断服务程序有下面操作。

<更改周期(CR000)>
1. TM00 和 CR000 (TOC001 = 0)相等时禁止定时器输出反转操作。
2. 禁止 INTTM000 中断(TMMK000 = 1)。
3. 重写 CR000。
4. 等待一个 TM00 计数时钟周期。
5. TM00 和 CR000(TOC001=1)相等时允许定时器输出反转操作。
6. INTTM000 (TMIF000 = 0)的中断请求标志位清零。
7. 允许 INTTM000 中断(TMMK000 = 0)。

<更改占空比(CR010)>
1. TM00 和 CR010 (TOC004 = 0)相等时禁止定时器输出反转操作。
2. 禁止 INTTM000 中断(TMMK000 = 1。
3. 重置 CR010。
4. 等待一个 TM00 计数时钟周期。
5. TM00 和 CR010(TOC004=1)相等时允许定时器输出反转操作。
6. INTTM000 (TMIF000 = 0)的中断请求标志位清零。
7. 允许 INTTM000 中断(TMMK000 = 0)。

当中断和定时器输出反转被禁止时（上面 1-4 步），定时器计数仍然继续。如果设置的 CR0n0 的值比较小，TM00 的值可能超过 CR0n0。因此设置 CR0n0 的值时要考虑当产生 INTTM000 中断后，定时器时钟和 CPU 时钟的影响。

备注　n = 0, 1

〈2〉 如果在定时器计数未执行上面的过程〈1〉期间改变 CR010 时，那么 CR010 中的值将被重新写入两次或更多，每次写入将引起 TO00 引脚电平的反转。

(18) 脉冲沿检测

★ 〈1〉 按照如下的方法，注意检测 TI0n0 引脚有效沿。
(a) 系统复位后，如果 TI0n0 引脚输入高电平，那么 16 位定时器计数器 00 (TM00) 的操作允许。
→如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。
(b) 如果当 TI0n0 引脚是高电平的时候，停止 TM00 操作，TI0n0 引脚的输入变低后 TM00 操作允许。
→ 如果下降沿或双沿被指定为 TI0n0 引脚的有效沿，，TM00 操作允许后，马上会检测到一个下降沿
(c) 如果当 TI0n0 引脚是低电平的时候，停止 TM00 操作，TI0n0 引脚的输入变高后 TM00 操作允许。
→ 如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。

备注　n = 0, 1

〈2〉 当 TI000 有效沿用作计数时钟和用作捕捉触发时，所用的消除噪音的采样时钟是不同的。在前一种情况中，计数时钟为 f_X。而在后一种情况中，由预分频模式寄存器 00(PRM00)选择计数时钟。直到采样有效沿并且有效电平被检测到两次之后才能完成捕捉操作，因此，可以消除短脉冲宽度噪音。

(19) 外部事件计数器

要想知道外部事件计数器的计数值，应该去读 TM00。

(20) PPG 输出

〈1〉 CR000 和 CR010 中设置的值应在如下范围内:
0000H < CR010 < CR000 ≤ FFFFH (禁止设置 CR000 为 0000H)
〈2〉 通过 PPG 输出产生的脉冲的周期（CR000 的值 + 1）的占空比是：(CR010 的值+ 1)/ (CR000 的值 + 1)。

(21) 停止模式或系统时钟停止模式的设置

除 TI000 引脚有效沿作为计数器时钟信号输入以外，在设置 STOP 模式或系统时钟停止模式前停止定时器工作；否则在系统时钟启动时，定时器将发生误差。

(22) P31/TI010/TO00 引脚

当 P31 用做 TI010 的有效沿输入引脚时,不能用作定时器输出引脚(TO00)。同样，当 P31 被用于定时器输出引脚 TO00 时，不能用于 TI010 的有效沿输入引脚。

第七章　　8 位定时器 80

7.1　8 位定时器 80 的功能

8 位定时器 80 有 8 位间隔定时器的功能，并可按预先设定的时间间隔产生中断。

表 7-1. 8 位定时器 80 的间隔时间

	最小间隔时间	最大间隔时间	分辨率
f_{XP} = 8.0 MHz	$2^6/f_{XP}$ (8 μs)	$2^{14}/f_{XP}$ (2.05 ms)	$2^6/f_{XP}$ (8 μs)
	$2^8/f_{XP}$ (32 μs)	$2^{16}/f_{XP}$ (8.19 ms)	$2^8/f_{XP}$ (32 μs)
	$2^{10}/f_{XP}$ (128 μs)	$2^{18}/f_{XP}$ (32.7 ms)	$2^{10}/f_{XP}$ (128 μs)
	$2^{16}/f_{XP}$ (8.19 ms)	$2^{24}/f_{XP}$ (2.01 s)	$2^{16}/f_{XP}$ (8.19 ms)
f_{XP} = 10.0 MHz	$2^6/f_{XP}$ (6.4 μs)	$2^{14}/f_{XP}$ (1.64 ms)	$2^6/f_{XP}$ (6.4 μs)
	$2^8/f_{XP}$ (25.6 μs)	$2^{16}/f_{XP}$ (6.55 ms)	$2^8/f_{XP}$ (25.6 μs)
	$2^{10}/f_{XP}$ (102 μs)	$2^{18}/f_{XP}$ (26.2 ms)	$2^{10}/f_{XP}$ (102 μs)
	$2^{16}/f_{XP}$ (6.55 ms)	$2^{24}/f_{XP}$ (1.68 s)	$2^{16}/f_{XP}$ (6.55 ms)

备注　　f_{XP}: 外围硬件时钟的振荡频率。

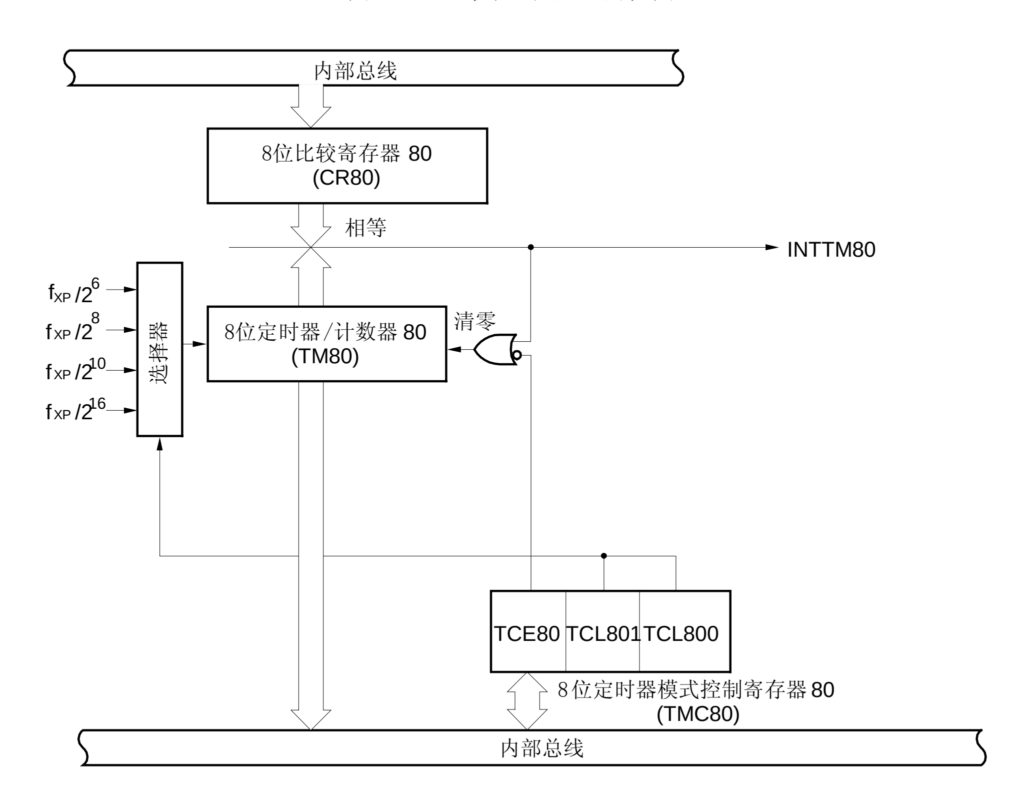
7.2　8 位定时器 80 的配置

8 位定时器 80 由以下硬件组成。

表 7-2.　8 位定时器 80 的配置

项目	配置
定时器计数器	8 位定时器计数器 80 (TM80)
寄存器	8 位比较寄存器 80 (CR80)
控制寄存器	8 位定时器模式控制寄存器 80 (TMC80)

图 7-1.　8 位定时器 80 的框图



备注　　f_{XP}: 外围硬件时钟的振荡频率。

(1) 8 位比较寄存器 80 (CR80)

8 位比较寄存器 80 用来将其设定值与 8 位定时器/计数器 80 的值进行比较，当 8 位比较寄存器 80 的值与 8 位定时器/计数器 80 的值相等时，产生中断请求信号（INTTM80）。

CR80 可通过一个 8 位的存储操作指令来设置。其值可为 00H～FFH。

复位后寄存器值不确定。

图 7-2. 8 位比较寄存器 80（CR80）格式

地址：FFCDH　　复位后：不确定　　只写

符号	7	6	5	4	3	2	1	0
CR80								

注意事项　在定时器操作期间，不能改变 CR80 的值。如果在定时器操作期间改变 CR80 的值，则立刻产生中断请求信号。

(2) 8 位定时器计数器 80 (TM80)

该寄存器用来对计数脉冲进行计数。

TM80 的值可通过 8 位存储操作指令进行读取。

复位后 TM80 的值被清零。

图 7-3.　8 位定时器计数器 80（TM80）的格式

地址：FFCEH　　复位后：00H　　只读

符号	7	6	5	4	3	2	1	0
TM80								

7.3　控制 8 位定时器 80 的寄存器

8 位定时器 80 可通过 8 位定时器模式控制寄存器 80（TMC80）进行控制。

(1) 8 位定时器模式控制寄存器 80 (TMC80)

TMC80 用来允许或停止 8 位定时器/计数器 80（TM80）的操作，并可用来设置 TM80 的计数时钟。
该寄存器可用 1 位或 8 位存储操作指令进行设置。
复位寄存器值清零。

图 7-4.　8 位定时器模式控制寄存器 80（TMC80）的格式

地址:　FFCCH　　复位后:　00H　　R/W

符号	<7>	6	5	4	3	2	1	0
TMC80	TCE80	0	0	0	0	TCL801	TCL800	0

TCE80	TM80 的控制操作
0	停止操作（ TM80 被清零）
1	允许操作

TCL801	TCL800	8 位定时器 80 的计数时钟选择		
			f_{XP} = 8.0 MHz	f_{XP} = 10.0 MHz
0	0	$f_{XP}/2^6$	125 kHz	156.3 kHz
0	1	$f_{XP}/2^8$	31.25 kHz	39.06 kHz
1	0	$f_{XP}/2^{10}$	7.81 kHz	9.77 kHz
1	1	$f_{XP}/2^{16}$	0.12 kHz	0.15 kHz

注意事项　1. 定时器操作停止后必须对 TMC80 重新设置。
2. 第 0 位和第 6 位必须清零。

备注　f_{XP}: 外围硬件时钟的振荡频率。

7.4　8 位定时器 80 的操作

7.4.1　作为间隔定时器的操作

当 8 位定时器 80 作为间隔定时器时，可以按照 8 位比较寄存器（CR80）中预先设置的计数值为间隔重复产生中断。

将 8 位定时器 80 用作间隔定时器的步骤如下：

〈1〉　禁止操作 8 位定时器计数器 80（将 8 位定时器模式控制寄存器 80（TMC80）的第 7 位（TCE80）设置为 0）。

〈2〉　设置 8 位定时器 80 的计数时钟（参见表 7－3 和 7－4）。

〈3〉　设置 CR80 的计数值。

〈4〉　允许 TM80 的操作（设置 TCE80 为 1）。

当 8 位定时器计数器 80 的计数值等于 CR80 的设置值时，TM80 的值被清零然后继续计数，同时产生中断请求信号 INTTM80。

表 7-3 和表 7-4 列出了间隔时间，图 7-5 表示间隔定时器操作的时序。

注意事项　1. 在定时器工作期间，不能改变 CR80 的值。如果在定时器工作期间改变 CR80 的值，则立刻产生相等中断请求信号。

2. 当使用 8 位存储操作指令设置 TMC80 的计数时钟并同时允许 TM80 时，则在定时器启动后一个周期内的误差可能是一个或多个时钟。因此，要将 TM80 作为间隔定时器使用，必须按上述顺序进行设置。

表 7-3.　8 位定时器 80 的间隔时间（f_{XP} = 8.0 MHz）

TCL801	TCL800	最小间隔时间	最大间隔时间	分辨率
0	0	$2^6/f_{XP}$（8 μs）	$2^{14}/f_{XP}$（2.05 ms）	$2^6/f_{XP}$（8 μs）
0	1	$2^8/f_{XP}$（32 μs）	$2^{16}/f_{XP}$（8.19 ms）	$2^8/f_{XP}$（32 μs）
1	0	$2^{10}/f_{XP}$（128 μs）	$2^{18}/f_{XP}$（32.7 ms）	$2^{10}/f_{XP}$（128 μs）
1	1	$2^{16}/f_{XP}$（8.19 ms）	$2^{24}/f_{XP}$（2.01 s）	$2^{16}/f_{XP}$（8.19 ms）

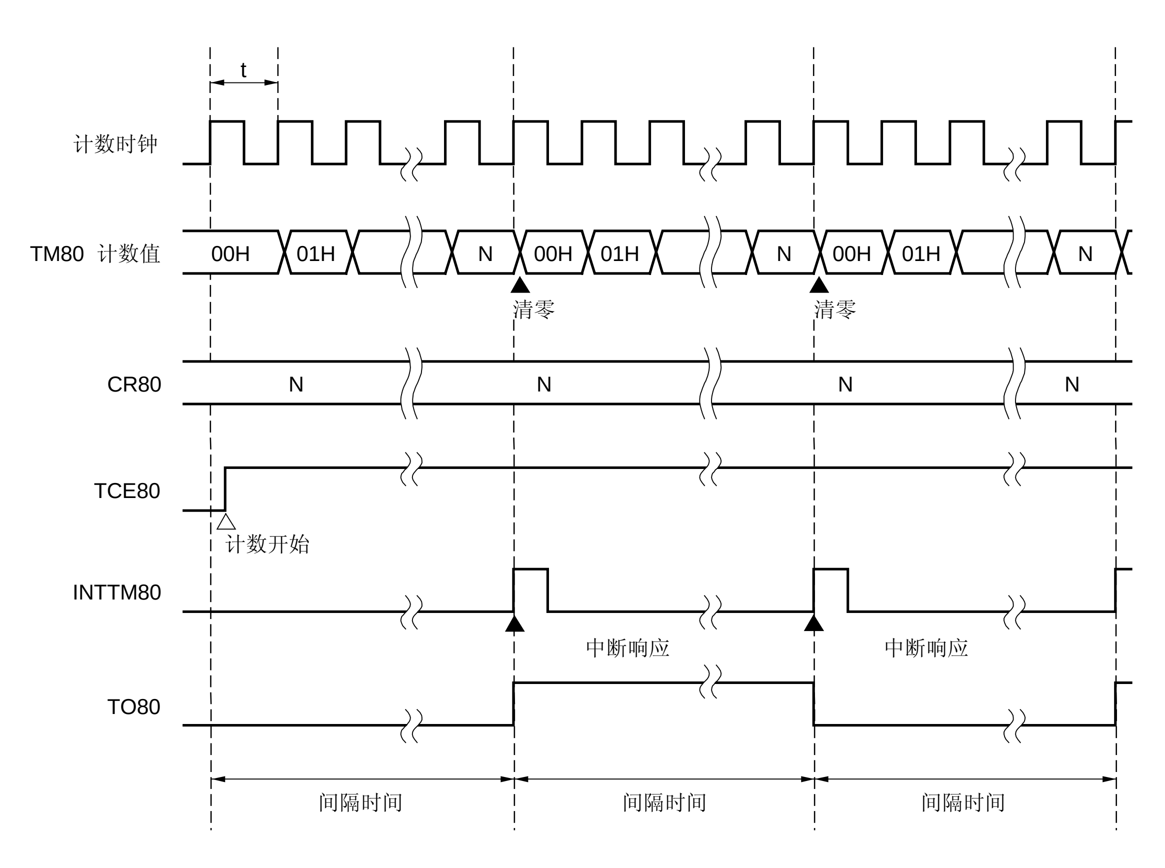
备注　　f_{XP}：外围硬件时钟的振荡频率。

表 7-4.　8 位定时器 80 的间隔时间（f_{XP} = 10.0 MHz）

TCL801	TCL800	最小间隔时间	最大间隔时间	分辨率
0	0	$2^6/f_{XP}$（6.4 μs）	$2^{14}/f_{XP}$（1.64 ms）	$2^6/f_{XP}$（6.4 μs）
0	1	$2^8/f_{XP}$（25.6 μs）	$2^{16}/f_{XP}$（6.55 ms）	$2^8/f_{XP}$（25.6 μs）
1	0	$2^{10}/f_{XP}$（102 μs）	$2^{18}/f_{XP}$（26.2 ms）	$2^{10}/f_{XP}$（102 μs）
1	1	$2^{16}/f_{XP}$（6.55 ms）	$2^{24}/f_{XP}$（1.68 s）	$2^{16}/f_{XP}$（6.55 ms）

备注　　f_{XP}：外围硬件时钟的振荡频率。

图 7-5.　间隔定时器操作的时序图



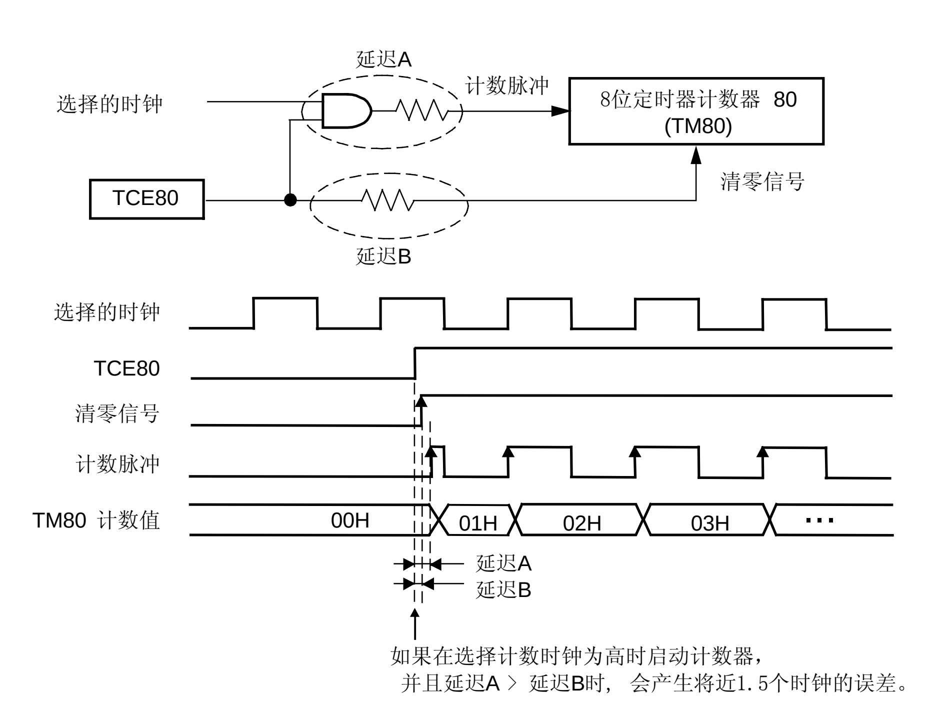
备注　　间隔时间 = (N + 1) × t：N = 00H ～ FFH

7.5　8 位定时器 80 的注意事项

(1) 定时器启动后的误差

从启动定时器到产生中断信号的时间包含了一个将近 1.5 个时钟周期的误差。这是因为如果在计数时钟为高电平时启动定时器，那么上升沿可能被立即检测到，并且计数器累加（参看图 7－6）。

图 7-6.　1.5 个（最大）时钟周期误差的情形



(2) 8 位比较寄存器 80 的设置

8 位比较寄存器 80（CR80）可设置为 00H。

(3) 设置为 STOP 模式时注意事项

在执行 STOP 指令之前必须先停止定时器操作（TCE80=0）。

第八章　8位定时器 H1

8.1　8 位定时器 H1 的功能

8位定时器 H1 的功能如下：

- 间隔定时器
- PWM 输出模式
- 方波输出

8.2　8 位定时器 H1 的配置

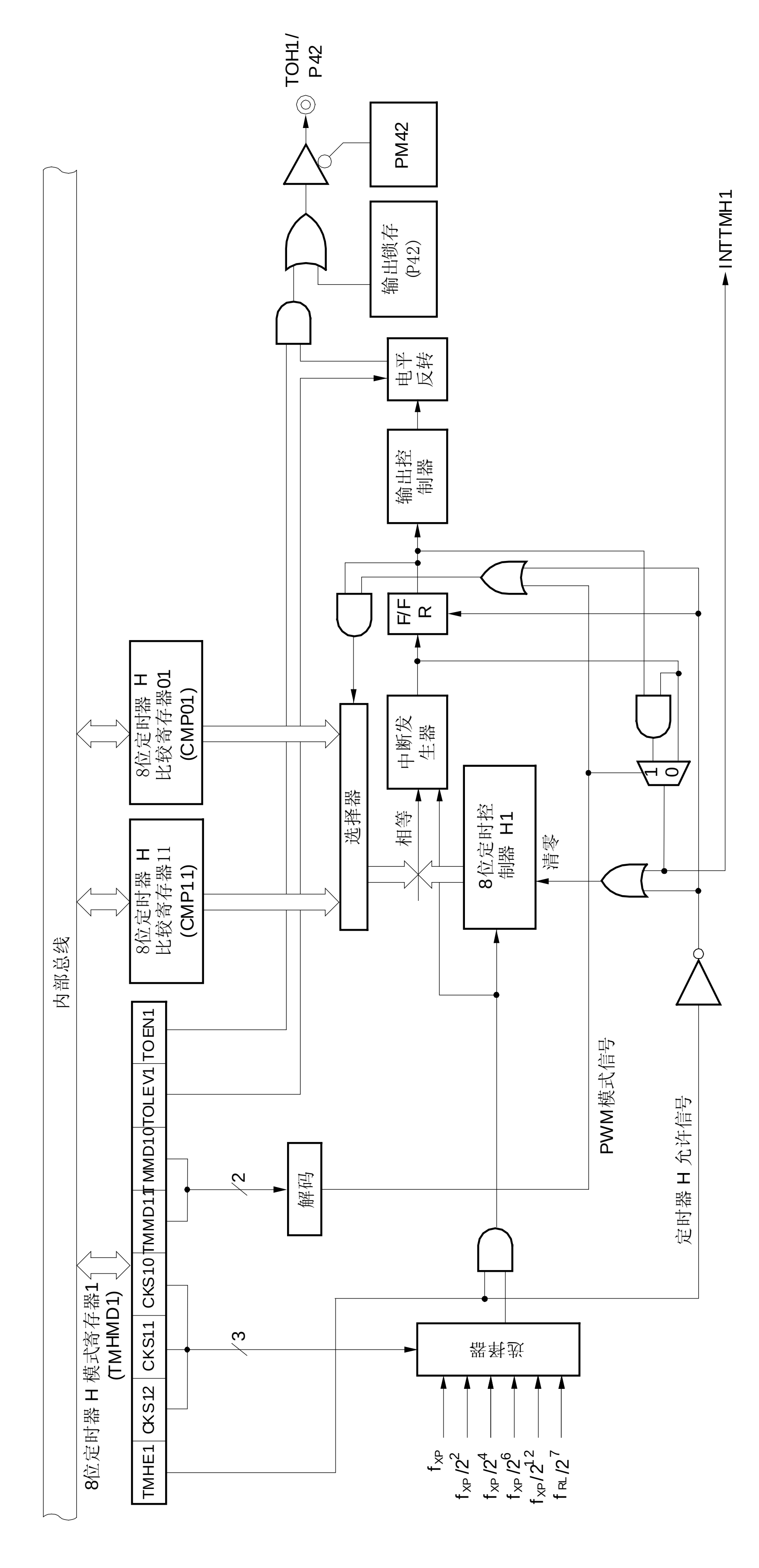
8位定时器 H1 由以下硬件组成。

表 8-1.　8位定时器 H1 的配置

项目	配置
定时器寄存器	8位定时器计数器 H1
寄存器	8位定时器H比较寄存器01 (CMP01) 8位定时器H比较寄存器11 (CMP11)
定时器输出	TOH1
控制寄存器	8位定时器H模式寄存器1 (TMHMD1) 端口模式寄存器4 (PM4) 端口寄存器4 (P4)

图8-1 是结构框图。

图 8-1.　8 位定时器 H1 的框图



(1) 8 位定时器 H 比较寄存器 01 (CMP01)

这个寄存器可通过 8 位存储器操作指令进行读写。

复位后寄存器被清零。

图 8-2.　8 位定时器 H 比较寄存器 01（CMP01）的格式

地址: FF0EH　复位后: 00H　读/写

符号	7	6	5	4	3	2	1	0
CMP01								

注意事项　在定时器计数期间不能修改 CMP01。

(2) 8 位定时器 H 比较寄存器 11 (CMP11)

这个寄存器可通过 8 位存储器操作指令进行读写。

复位后寄存器值被清零。

图 8-3.　8 位定时器 H 比较寄存器 11（CMP11）的格式

地址: FF0FH　复位后: 00H　读/写

符号	7	6	5	4	3	2	1	0
CMP11								

CMP11 在定时器计数期间可被重写。

如果在定时器操作期间重写 CMP11 值，当计数值和修改以前的比较值一致时，修改后的比较值才生效。如果计数值和修改以前的比较值一致时 CPU 正向 CMP11 写入新值，则在下一个计数值和修改以前的比较值一致时修改后的比较值生效。

注意事项　在 PWM 输出模式中，当定时器停止计数后（TMHE1=0），再次启动计数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使前后设置的值相同也必须对 CMP11 重新设置）。

8.3　控制 8 位定时器 H1 的寄存器

以下三个寄存器用来控制 8 位定时器 H1。

- 8 位定时器 H 模式寄存器 1 (TMHMD1)
- 端口模式寄存器 4 (PM4)
- 端口寄存器 4 (P4)

(1) 8 位定时器 H 模式寄存器 1 (TMHMD1)

该寄存器控制 8 位定时器 H 的模式。

该寄存器可通过 1 位或 8 位存储器操作指令进行设置。

复位后寄存器值被清零（00H）。

图 8-4.　8 位定时器 H 模式寄存器 1（TMHMD1）的格式

地址: FF70H　复位后:　00H　读/写

符号	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	定时器操作使能
0	停止定时器计数操作(计数器清零)
1	允许定时器计数操作 (通过输入时钟启动计数操作)

CKS12	CKS11	CKS10	计数时钟(f_{CNT}) 选择
0	0	0	f_{XP} (10 MHz)
0	0	1	$f_{XP}/2^2$ (2.5 MHz)
0	1	0	$f_{XP}/2^4$ (625 kHz)
0	1	1	$f_{XP}/2^6$ (156.25 kHz)
1	0	0	$f_{XP}/2^{12}$ (2.44 kHz)
1	0	1	$f_{RL}/2^7$ (1.88 kHz (TYP.))
其它			禁止设置

TMMD11	TMMD10	定时器操作模式
0	0	间隔定时器模式
1	0	PWM 输出模式
其它		禁止设置

TOLEV1	定时器输出电平控制(在默认的模式中)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	不允许输出
1	允许输出

注意事项　1. 当 TMHE1 设置为 1 时，禁止对 TMHMD1 寄存器的其它位进行设置。

2. 在 PWM 输出模式下，当定时器停止计数后（TMHE1=0），再次启动计数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使前后设置的值相同也必须对 CMP11 重新设置）。

备注　1. f_{XP}：外围硬件设备时钟振荡频率

2. f_{RL}：低速 Ring-OSC 时钟振荡频率

3. 上表中括号内的值对应于 f_{XP}＝10MHz，f_{RL}=240kHz（典型值）。

(2) 端口模式寄存器 4 (PM4)

该寄存器可以位选端口 4 的输入输出模式。

当使用 P42/TOH1 引脚作为定时器输出时，要对 PM42 进行清零并将 P42 的输出锁存清零。

PM4 可通过 1 位或 8 位存储器操作指令进行设置。

复位后寄存器值为 FFH。

图 8-5.　端口模式寄存器 4（PM4）的格式

地址: FF24H　　复位后: FFH　　R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	P4n 引脚 I/O 模式选择 (n = 0 ～ 7)
0	输出模式（输出缓冲开启）
1	输入模式（输出缓冲关闭）

8.4　8 位定时器 H1 的操作

8.4.1　作为间隔定时器/方波输出的操作

当 8 位定时器计数器 H1 与比较寄存器 01（CMP01）相等时，产生中断请求信号 INTTMH1，8 位定时器计数器 H1 被清零。

在间隔定时器模式下不使用比较寄存器 11（CMP11）。因为即使设置 CMP11 寄存器，也不会对 8 位定时器计数器 H1 和 CMP11 寄存器的相等进行检测，所以，不影响定时器的输出。

当设置定时器 H 模式寄存器 1（TMHMD1）的第 0 位（TOEN1）为 1 时，从 TOH1 可输出任意频率的方波（占空比为 50 %）。

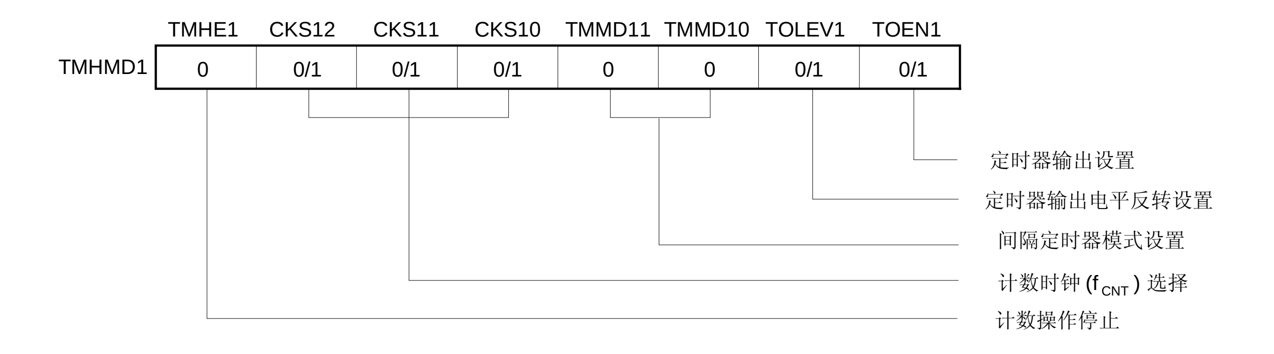
(1) 用法

以相同时间间隔重复产生 INTTMH1 信号。

〈1〉　各个寄存器的设置

图 8-6.　间隔定时器/方波输出操作时的寄存器设置

(i) 定时器 H 模式寄存器 1（TMHMD1）的设置



(ii)　CMP01 寄存器设置

- 比较值(N)

〈2〉　当 TMHE1=1 开始计数操作。

〈3〉　当 8 位定时器计数器 H1 的值等于 CMP01 寄存器的值时，产生中断请求信号 INTTMH1，8 位定时器计数器 H1 被清零。

间隔时间= $(N+1)/f_{CNT}$

〈4〉　此后，以相同的时间间隔产生 INTTMH1 中断。要停止计数操作，将 TMHE1 清零即可。

(2) 时序图

间隔定时器/方波输出操作的时序图如下：

图 8-7. 间隔定时器/方波输出操作的时序图(1/2)

(a) 基本操作

〈1〉 将 TMHE1 位设置为 1，允许计数操作。操作允许后，在不到 1 个时钟周期的时间内计数时钟开始计数。

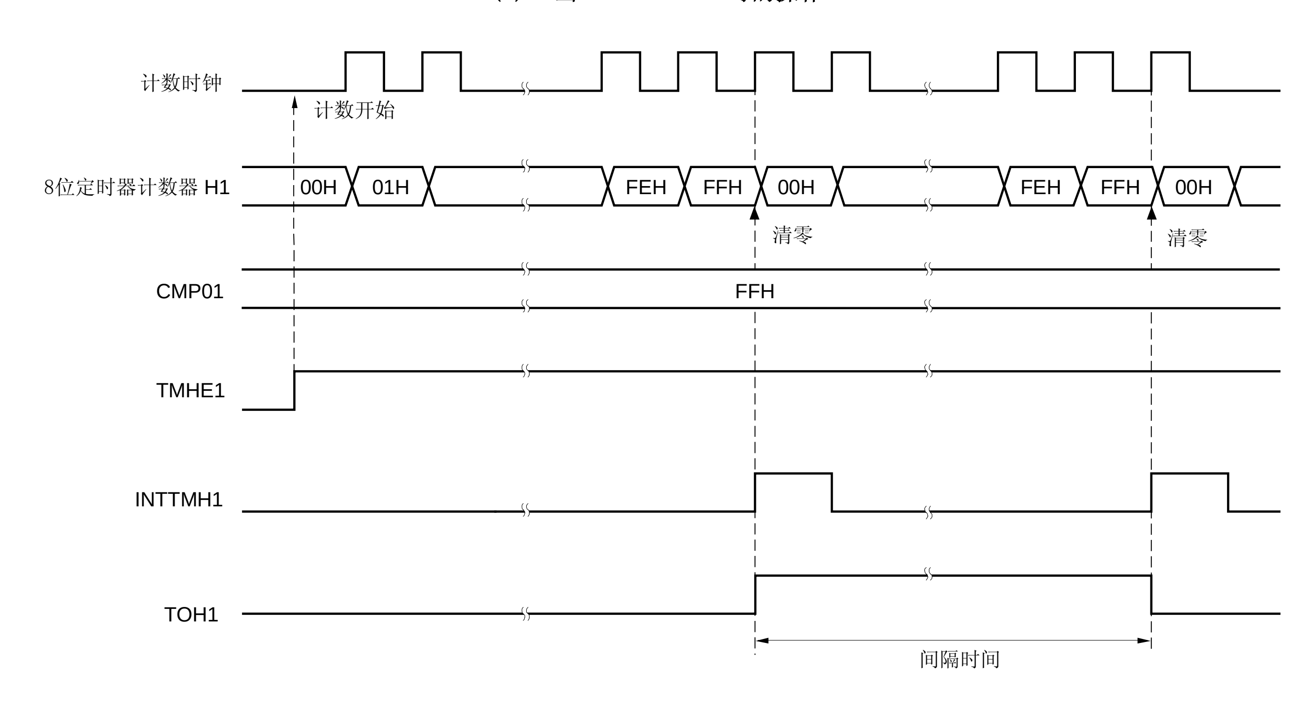
〈2〉 当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时， 8 位定时器计数器 H1 的值清零， TOH1 输出电平反转并输出 INTTMH1 信号。

〈3〉 在定时器 H1 操作期间将 TMHE1 位清零，可以使 INTTMH1 信号和 TOH1 输出无效。如果一旦没有 INTTMH1 信号和 TOH1 输出，则电平保持不变。

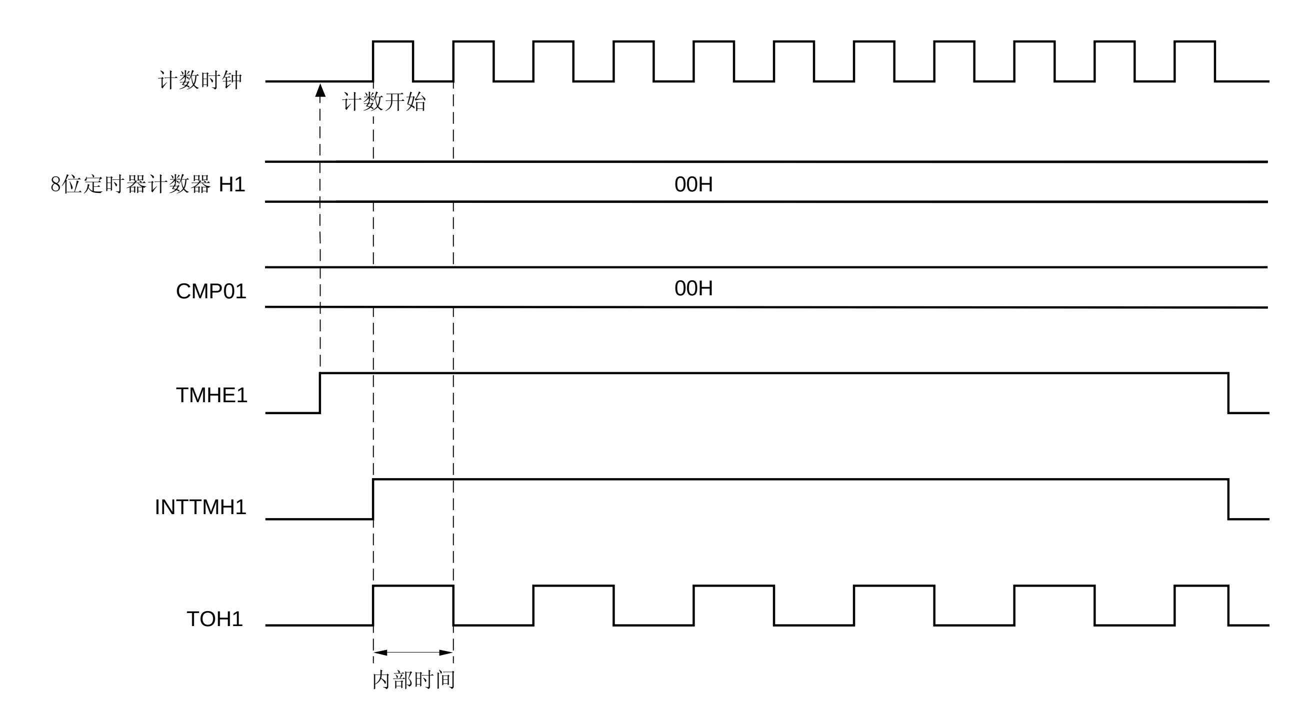
备注 N = 01H ～ FEH

图 8-7.　间隔定时器/方波输出操作的时序 (2/2)

(b)　当 CMP01 = FFH 时的操作



(c)　当 CMP01 = 00H 时的操作



8.4.2 用作 PWM 输出模式的操作

在 PWM 输出模式中，可输出具有任意占空比和周期的脉冲。

8 位定时器比较寄存器 01(CMP01)用来控制定时器输出(TOH1)的周期。禁止在定时器操作期间重写 CMP01 寄存器。

8 位定时器比较寄存器 11(CMP11)用来控制定时器输出(TOH1)的占空比。允许在定时器操作期间重写 CMP11 寄存器。

PWM 输出模式中的操作如下所示。

在定时器开始计数后当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，TOH1 开始输出且 8 位定时器计数器 H1 清零。当 8 位定时器计数器 H1 与 CMP11 寄存器的值相等时，TOH1 停止输出。

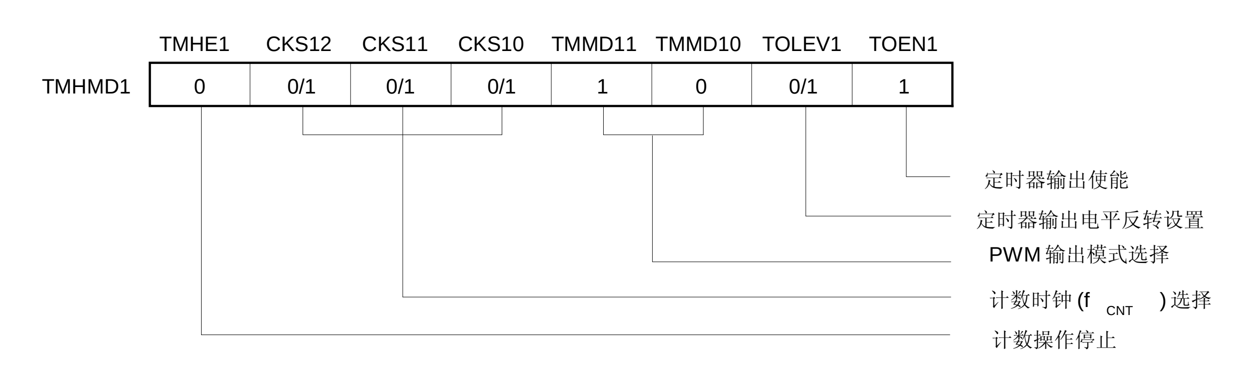
(1) 用法

在 PWM 输出模式中，可输出具有任意占空比和周期的脉冲。

〈1〉 设置寄存器

图 8-8.　PWM 输出模式下的寄存器设置

(i) 设置定时器 H 模式寄存器 1 (TMHMD1)



(ii) 设置 CMP01 寄存器

- 比较值(N): 周期设置

(iii) 设置 CMP11 寄存器

- 比较值(M): 占空比设置

备注　00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH

〈2〉 当 TMHE1 = 1 时开始计数。

〈3〉 CMP01 是在允许计数器操作后首先用于比较的比较寄存器。当 8 位定时器计数器 H1 与 CMP01 寄存器的值相等时，8 位定时器计数器 H1 清零、产生中断请求信号(INTTMH1)并且 TOH1 开始输出。同时切换比较寄存器（与 8 位定时器比较的寄存器），由 CMP01 改为 CMP11。

〈4〉 当8 位定时器计数器H1 与CMP11 寄存器相等时，停止TOH1 输出，并切换比较寄存器（与8 位定时器比较的寄存器）：由CMP11 改为CMP01 。此时不对8 位定时器计数器清零，也不产生INTTMH1 信号。

〈5〉 通过重复执行过程〈3〉 和〈4〉，可获取具有任意占空比的脉冲。

〈6〉 若要停止计数操作，设置 TMHE1 = 0。

若 CMP01 的值是 N，CMP11 的值是 M，计数时钟是 f_{CNT}，则 PWM 脉冲的输出周期和占空比如下所示。

PWM 脉冲输出周期 = (N+1)/f_{CNT} 占空比=有效宽度： PWM 的总宽度= (M + 1) : (N + 1)

注意事项 1. 在 PWM 输出模式下，在重写寄存器之后，需要 3 个操作周期（使用 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）来传送 CMP11 寄存器的值。

2. 当定时器停止计数后（TMHE1=0），再次启动计数操作时（TMHE1=1），必须对 CMP11 赋值（即使设置的是相同的值，也必须再次赋值）。

(2) 时序图

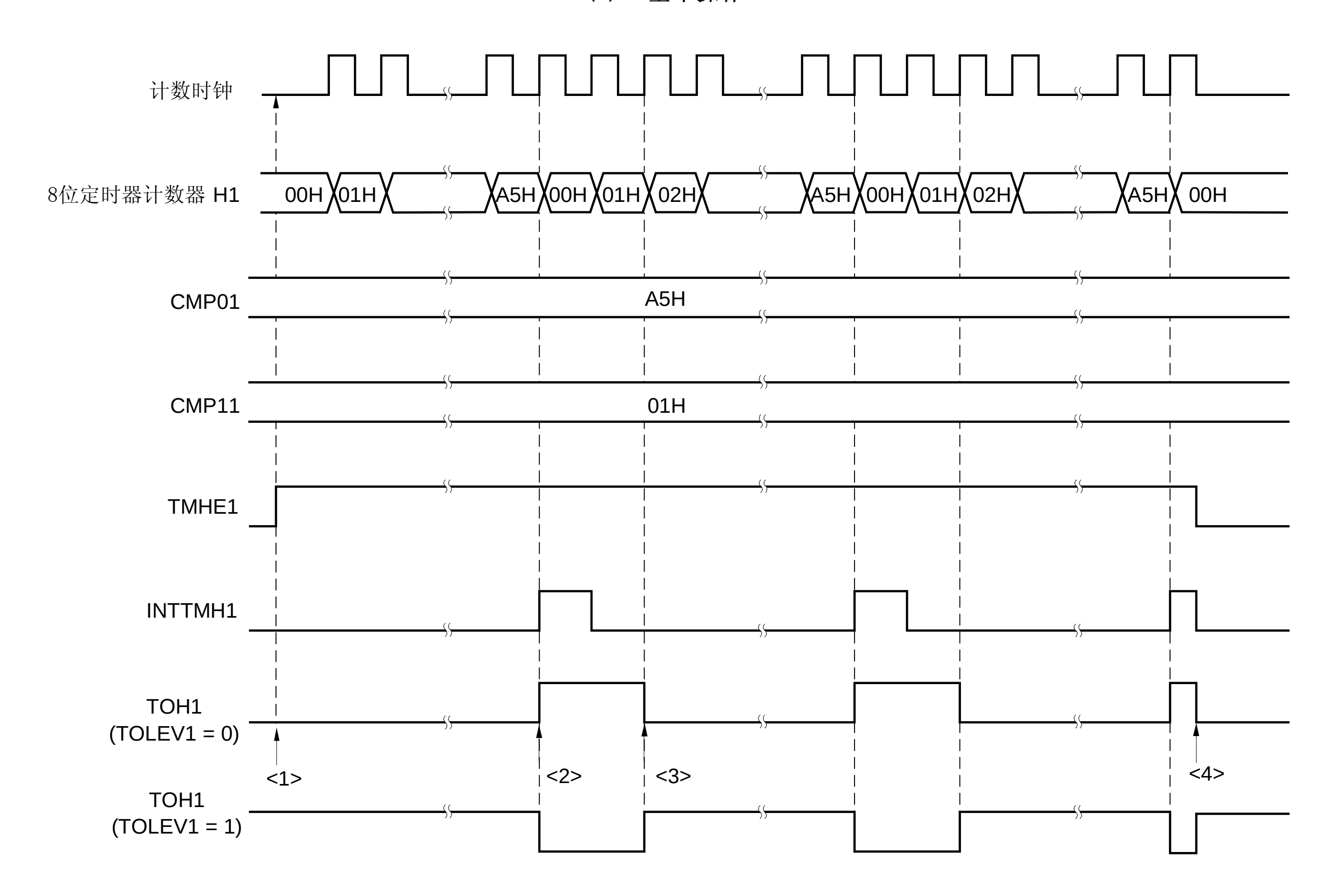
PWM 输出模式中的时序操作过程如下所示。

注意事项 CMP11 的取值（M）和 CMP01 的取值(N)应在如下范围内。

00H ≤ CMP11 (M) ＜ CMP01 (N) ≤ FFH

图 8-9.　PWM 输出模式下的时序操作(1/4)

(a)　基本操作



〈1〉 TMHE1 置 1，允许计数操作。屏蔽一个计数时钟，启动 8 位定时器计数器 H1。此时，TOH1 输出保持无效（TOLEV1=0）。

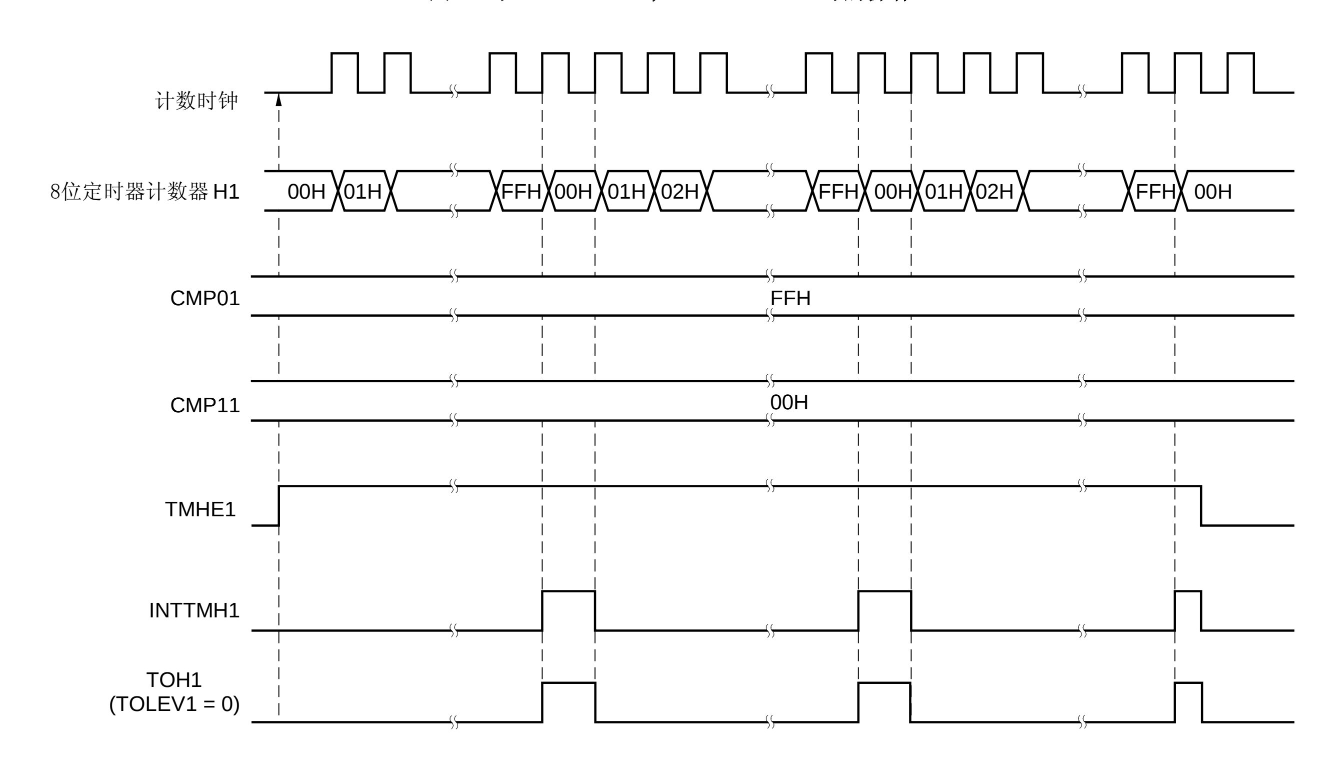
〈2〉 当 8 位定时器计数器 H1 和 CMP01 相等时，TOH1 输出电平反转，8 位定时器计数器 H1 清零，并输出 INTTMH1 中断信号。

〈3〉 当 8 位定时器计数器 H1 和 CMP11 相等时，TOH1 输出返回初始电平。而此时并不对 8 位定时器计数器 H1 清零，也不输出 INTTMH1 中断信号。

〈4〉 在定时器 H1 操作期间将 TMHE1 位清零，会使 INTTMH1 信号和 TOH1 输出无效。

图8-9.　PWM输出模式下的操作时序(2/4)

(b)　当CMP01 = FFH, CMP11 = 00H时的操作



(c)　当CMP01 = FFH, CMP11 = FEH时的操作

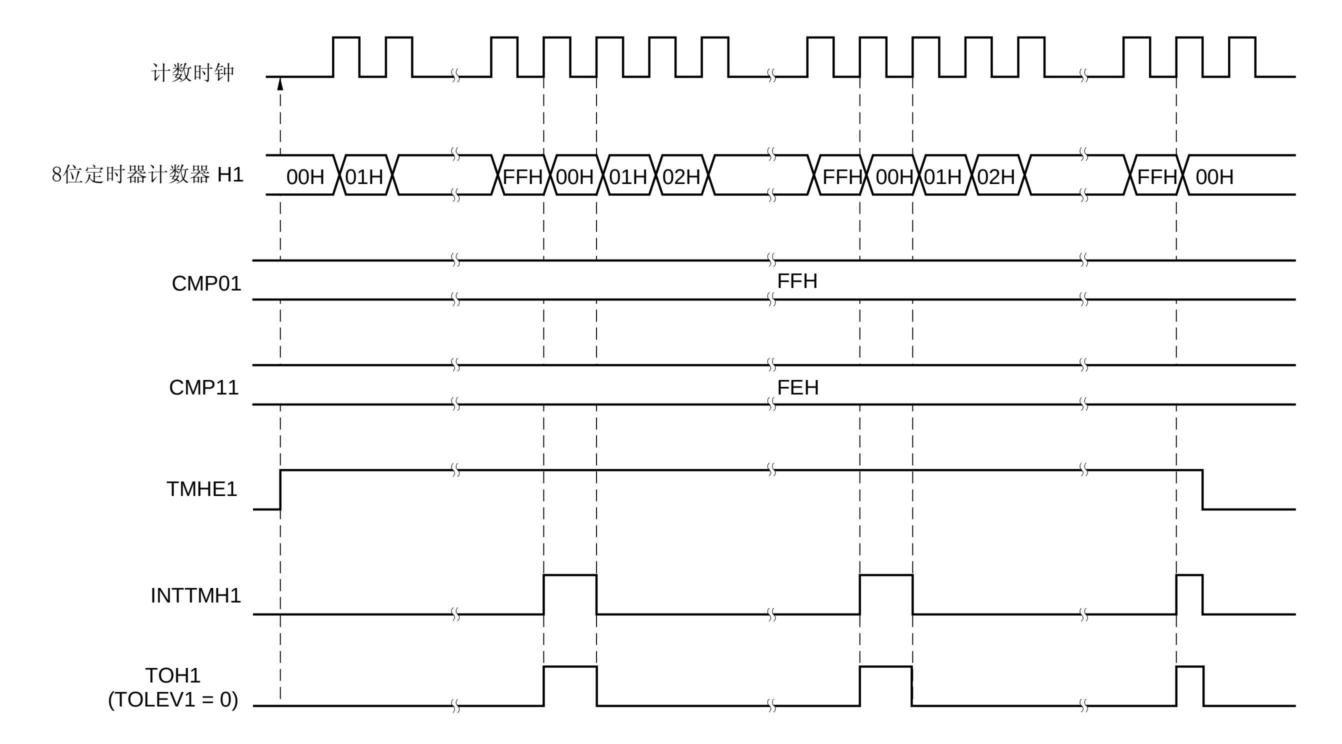


图 8-9.　PWM 输出模式下的操作时序（3/4）

（d）　当 CMP01 = 01H，CMP11 = 00H 时的操作

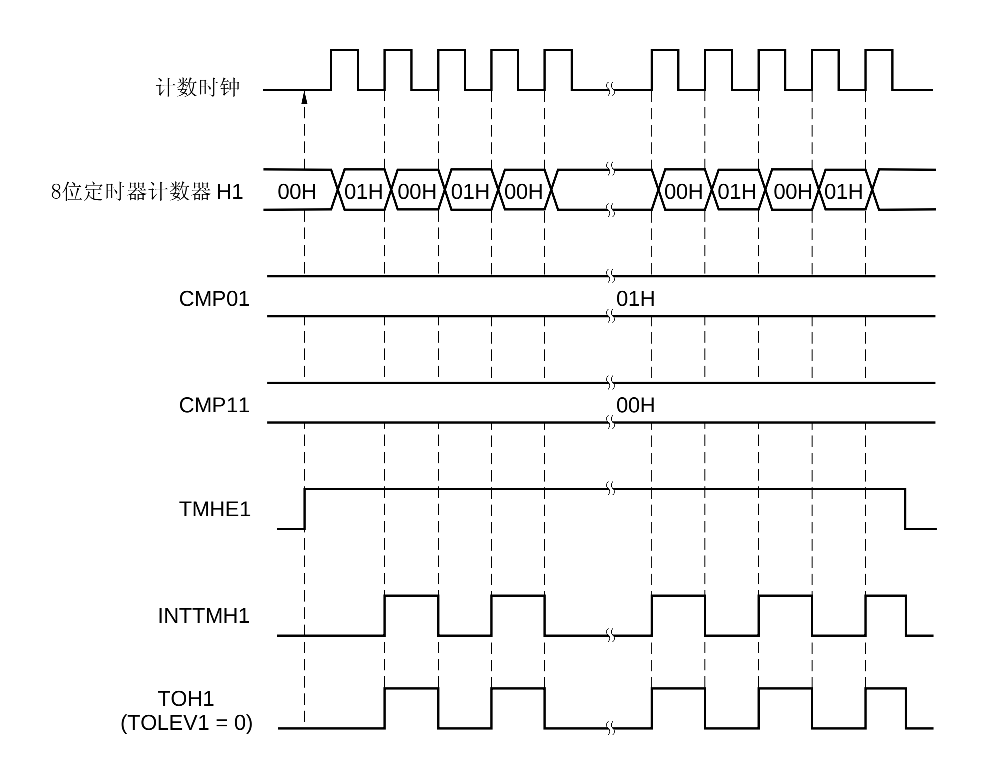
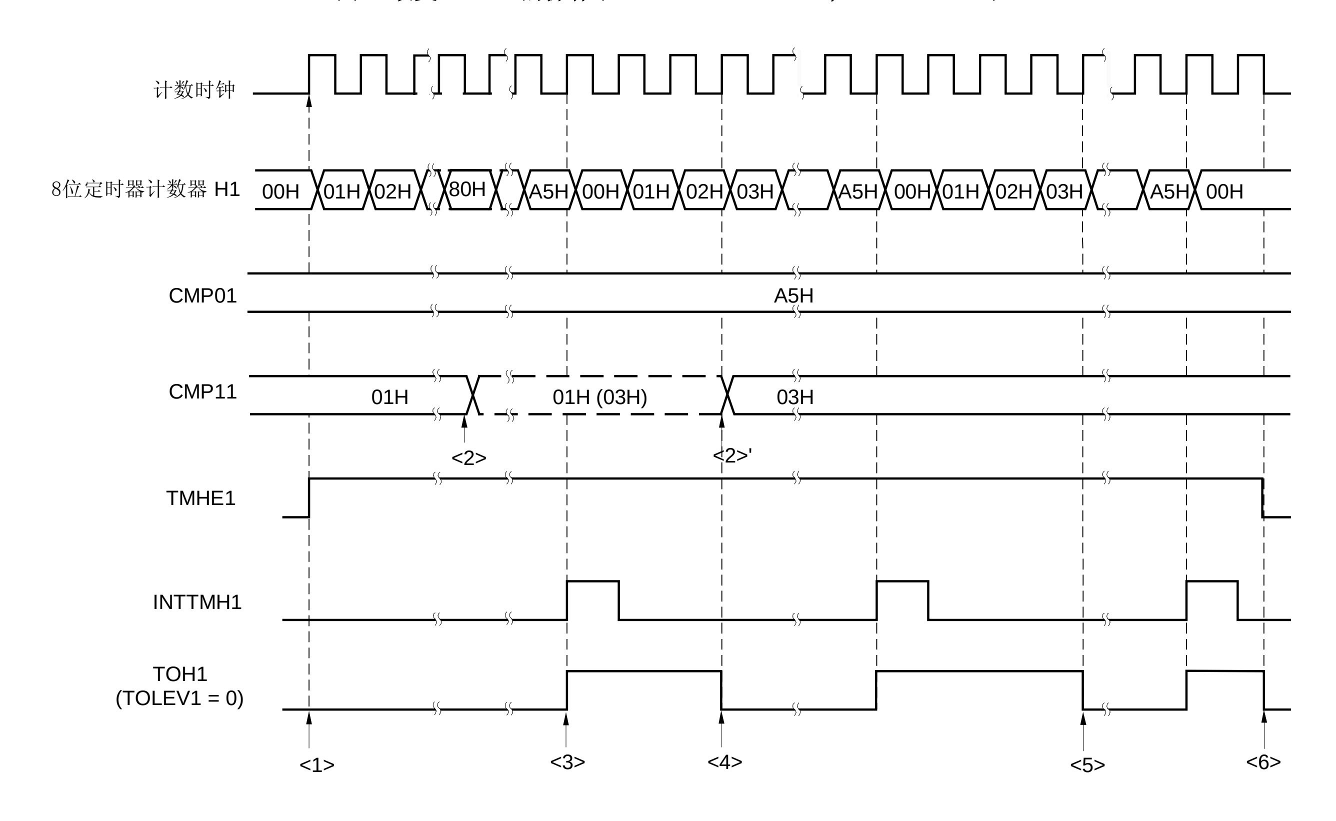


图 8-9.　PWM 输出模式下的操作时序(4/4)

(e)　改变 CMP11 的操作(CMP11 = 01H → 03H, CMP01 = A5H)



〈1〉　TMHE1 置 1，允许计数操作。屏蔽 1 个计数时钟，启动 8 位定时器计数器 H1。此时 TOH1 输出保持无效。（TOLEV1=0）

〈2〉　在定时器计数器操作期间可改变 CMP11 的值。该操作与计数时钟不同步。

〈3〉　当 8 位定时器计数器 H1 与 CMP01 的值相等时，8 位定时器计数器 H1 的值清零，TOH1 输出有效，输出 INTTMH1 信号。

〈4〉　如果 CMP11 的值改变，则改变后的值被锁存，且不传送到寄存器。当 8 位定时器计数器 H1 的值与修改前 CMP11 寄存器的值相等时，则将值传送至 CMP11，CMP11 的值改变(〈2〉)。
但从 CMP11 的值被改变到将值传送至寄存器需要 3 个或更多的计数时钟。如果在 3 个计数时钟内产生了一个相等信号，则不能将改变后的值传送至寄存器。

〈5〉　当 8 位定时器计数器 H1 的值与改变后的 CMP11 的值相等时，TOH1 输出无效。此时不对 8 位定时器计数器 H1 清零，也不产生 INTTMH1 信号。

〈6〉　在定时器 H1 操作期间将 TMHE1 位清零，会使 INTTMH1 信号和 TOH1 输出无效。

第九章　看门狗定时器

9.1　看门狗定时器的功能

看门狗定时器用于检测非预期的程序循环。如果检测到这样的一个程序循环，将产生一个内部复位信号。

如果看门狗定时器产生复位，则复位控制标志寄存器（RESF）的 第 4 位（WDTRF）置 1。需要了解 RESF 的详细信息，可参阅**第十五章 复位功能**。

表 9-1.　看门狗定时器的循环检测时间

循环检测时间	
低速 Ring-OSC 时钟操作期间	系统时钟操作期间
$2^{11}/f_{RL}$ (4.27 ms)	$2^{13}/f_X$ (819.2 μs)
$2^{12}/f_{RL}$ (8.53 ms)	$2^{14}/f_X$ (1.64 ms)
$2^{13}/f_{RL}$ (17.07 ms)	$2^{15}/f_X$ (3.28 ms)
$2^{14}/f_{RL}$ (34.13 ms)	$2^{16}/f_X$ (6.55 ms)
$2^{15}/f_{RL}$ (68.27 ms)	$2^{17}/f_X$ (13.11 ms)
$2^{16}/f_{RL}$ (136.53 ms)	$2^{18}/f_X$ (26.21 ms)
$2^{17}/f_{RL}$ (273.07 ms)	$2^{19}/f_X$ (52.43 ms)
$2^{18}/f_{RL}$ (546.13 ms)	$2^{20}/f_X$ (104.86 ms)

备注　1. f_{RL}：低速 Ring-OSC 时钟振荡频率。
2. f_X：　系统时钟振荡频率。
3. 图中括号里的值对应于 f_{RL} = 480 kHz (MAX.)，f_X = 10 MHz。

看门狗定时器的操作模式可根据片内低速 Ring-OSC 时钟的可选字节设置进行切换，如表 9-2 所示。

表 9-2.　可选字节的设置和看门狗定时器操作模式

	可选字节设置	
	低速 Ring-OSC 时钟不能被停止	低速 Ring-OSC 时钟可通过软件停止
看门狗定时器时钟源	固定为 f_{RL} 注 1	• 可由软件选择（f_X，f_{RL} 或停止） • 复位后：f_{RL}
复位后的操作	以最大间隔时间启动操作（$2^{18}/f_{RL}$）	以最大间隔时间启动操作　($2^{18}/f_{RL}$)
操作模式选择	间隔时间只能被修改 1 次	时钟选择/间隔时间只能被修改 1 次
特征	看门狗定时器不能被停止	看门狗定时器能被停止注 2

注　1.　在供电时不能停止低速 Ring-OSC 振荡器（复位期间除外）。

2.　如果看门狗定时器时钟源不同，则停止看门狗定时器的供应时钟所需的条件也不同。

〈1〉　如果时钟源是 f_X，　则在满足以下条件时看门狗定时器的供应时钟停止。

- 当 f_X 被停止。
- 在 HALT/STOP 模式下。
- 在振荡器稳定时间内。

〈2〉　如果时钟源是 f_{RL}，则在满足以下条件时看门狗定时器的供应时钟停止。

- 如果 CPU 时钟是 f_X 并且在执行 STOP 指令前由软件停止 f_{RL}。
- 在 HALT/STOP 模式下。

备注　1. f_{RL}：低速 Ring-OSC 时钟振荡频率

2. f_X：　系统时钟振荡频率

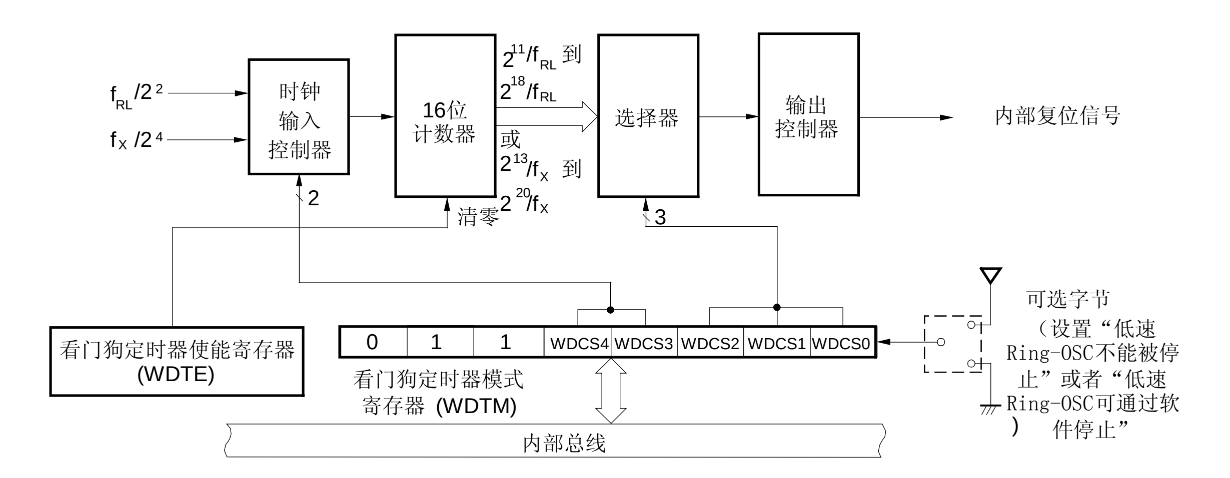
9.2　看门狗定时器的配置

看门狗定时器由以下硬件组成。

表　9-3.　看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器模式寄存器 (WDTM) 看门狗定时器使能寄存器 (WDTE)

图 9-1. 看门狗定时器的框图



9.3 控制看门狗定时器的寄存器

看门狗定时器由以下两种寄存器控制：

- 看门狗定时器模式寄存器 (WDTM)
- 看门狗定时器使能寄存器 (WDTE)

(1) 看门狗定时器模式寄存器 (WDTM)

该寄存器设置溢出时间和看门狗定时器的操作时钟。

该寄存器可由 8 位存储器操作指令设置， 并能被多次读取，但复位释放后只能被写 1 次。

复位输入将该寄存器置 67H。

图 9-2. 看门狗定时器模式寄存器(WDTM) 的格式

地址：FF48H　复位后：67H　读/写

符号	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4 注 1	WDCS3 注 1	操作时钟选择
0	0	低速 Ring-OSC 时钟 (f_{RL})
0	1	系统时钟 (f_X)
1	×	停止看门狗定时器

WDCS2 注 2	WDCS1 注 2	WDCS0 注 2	溢出时间设置	
			低速 Ring-OSC 时钟操作期间	系统时钟操作期间
0	0	0	$2^{11}/f_{RL}$ (4.27 ms)	$2^{13}/f_X$ (819.2 μs)
0	0	1	$2^{12}/f_{RL}$ (8.53 ms)	$2^{14}/f_X$ (1.64 ms)
0	1	0	$2^{13}/f_{RL}$ (17.07 ms)	$2^{15}/f_X$ (3.28 ms)
0	1	1	$2^{14}/f_{RL}$ (34.13 ms)	$2^{16}/f_X$ (6.55 ms)
1	0	0	$2^{15}/f_{RL}$ (68.27 ms)	$2^{17}/f_X$ (13.11 ms)
1	0	1	$2^{16}/f_{RL}$ (136.53 ms)	$2^{18}/f_X$ (26.21 ms)
1	1	0	$2^{17}/f_{RL}$ (273.07 ms)	$2^{19}/f_X$ (52.43 ms)
1	1	1	$2^{18}/f_{RL}$ (546.13 ms)	$2^{20}/f_X$ (104.86 ms)

注 1. 如果由可选字节指定为“低速 Ring-OSC 不能被停止”，则不能设置该项。 无论写入何值，都将选择低速 Ring-OSC 时钟。

2. 以最大周期复位释放 (WDCS2，1，0 = 1，1，1)。

注意事项 1. 第 7，6，5 位分别设置为 0，1 和 1（当通过可选字节选择“不能停止低速 Ring-OSC 时钟振荡器”，忽略其它值）。
2. 复位后，仅能由 8 位存储器操作指令对 WDTM 写入一次。如果试图写入第二次，就会产生内部复位信号。
3. 不能用 1 位存储器操作指令对 WDTM 进行操作。
4. 当使用自写入方式对 flash 存储器进行操作时，为看门狗设置足够的溢出时间（例如 一字节写入：最小 200 μs，一个块删除：最小 10 ms）。

备注 1. f_{RL}：低速 Ring-OSC 时钟振荡频率
2. f_X：系统时钟振荡频率
3. ×：不考虑
4. 图中括号里的值对应 f_{RL} = 480 kHz（MAX.），f_X = 10 MHz。

(2) 看门狗定时器使能寄存器（WDTE）

将 ACH 写入 WDTE 可对看门狗定时器计数器清零并再次启动计数操作。
可由 8 位存储器操作指令设置该寄存器。
复位输入将该寄存器置 9AH。

图 9-3. 看门狗定时器使能寄存器（WDTE）的格式

地址: FF49H　复位后: 9AH　读/写

符号	7	6	5	4	3	2	1	0
WDTE								

注意事项 1. 如果一个不是 ACH 的值被写入 WDTE，将产生一个内部复位信号。
2. 如果使用 1 位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。
3. 从 WDTE 读取的值是 9AH（与写入值（ACH）不同）。

9.4　看门狗定时器的操作

9.4.1　当由可选字节选择为 “低速Ring-OSC不能被停止”时的看门狗定时器操作

看门狗定时器的操作时钟固定为低速Ring-OSC时钟。

复位释放后，以最大周期（看门狗定时器模式寄存器（WDTM）的第 2，1，0 位都为 1（WDCS2，WDCS1，WDCS0）=1，1，1）启动操作，看门狗定时器操作不能被停止。

以下为复位释放后的看门狗定时器操作。

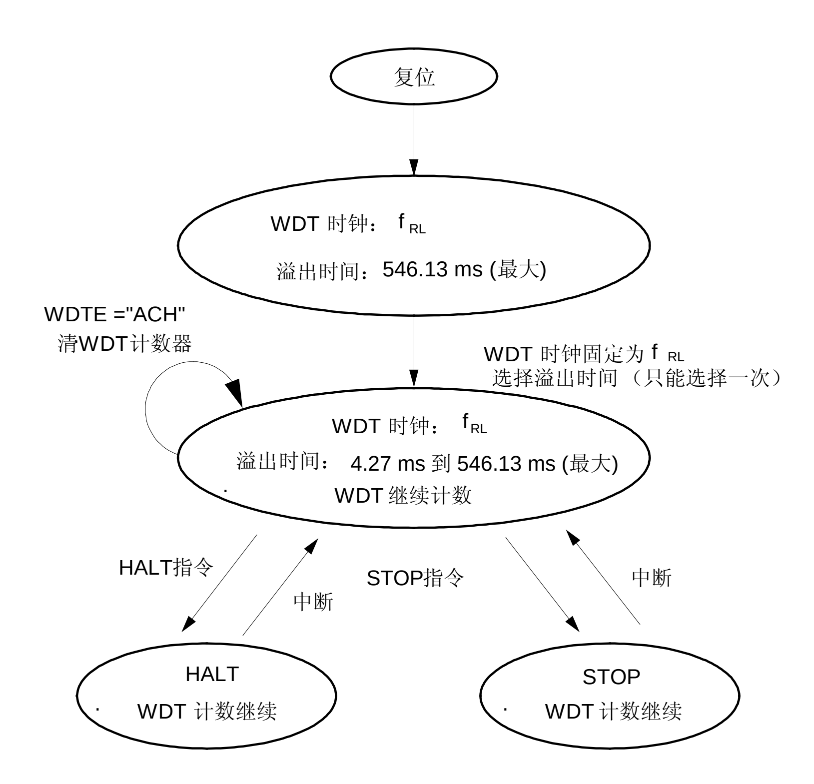
1. 复位释放后的状态如下。
   - 操作时钟：低速Ring-OSC时钟
   - 周期：$2^{18}/f_{RL}$ (546.13 ms：操作频率 f_{RL} = 480 kHz (最大值))
   - 开始计数
2. 由8位存储器操作指令对看门狗定时器模式寄存器（WDTM）中进行以下设置注1，2。
   - 周期：用第0位至第2位(WDCS2到 WDCS0)设置
3. 在执行上述过程后， 将ACH写入WDTE，这样计数器被清零，允许重新计数。

注　1. 不能改变操作时钟（低速Ring-OSC时钟）。任何写入WDTM第3位和第 4位 (WDCS3，WDCS4)的值将被忽略。
　　2. 只要对WDTM进行写入操作， 看门狗定时器计数器将被清零。

注意事项 **在这种模式中， 即使是执行 STOP 指令， 也不能停止看门狗定时器的操作。 对于 8 位定时器 H1 (TMH1)，可选择低速内部时钟的一个分频作为计数源，因此在 STOP 指令执行后， 看门狗定时器溢出之前， 可使用 TMH1 的中断请求对看门狗定时器清零。如果没有执行该过程，则在 STOP 指令执行后当看门狗定时器溢出时会产生内部复位信号。**

参考如下所示状态转换图。

图 9-4.　当由可选字节选择 “低速 Ring-OSC 不能被停止”时的状态转换图



9.4.2 当由可选字节选择为“低速 Ring-OSC 可通过软件停止”时的看门狗定时器操作

可选择看门狗定时器的操作时钟为低速 Ring-OSC 时钟或系统时钟。

复位释放后，以低速 Ring-OSC 时钟的最大周期（看门狗定时器模式寄存器（WDTM）的第 2 位，第 1 位和第 0 位 (WDCS2，WDCS1，WDCS0) = 1，1，1））启动操作。

以下为复位释放后的看门狗定时器操作：

1. 复位释放后的状态如下。
   - 操作时钟：低速 Ring-OSC 时钟
   - 周期：$2^{18}/f_{RL}$ (546.13 ms：操作频率 f_{RL} = 480 kHz (MAX.))
   - 开始计数
2. 由 8 位存储器操作指令对看门狗定时器模式寄存器 (WDTM) 进行以下设置[注 1，2，3]。
   - 操作时钟：由第 3 位和第 4 位 (WDCS3 and WDCS4)选择以下时钟。
     低速 Ring-OSC 时钟 (f_{RL})
     系统时钟 (f_X)
     停止看门狗定时器操作
   - 周期： 由第 2 位至第 0 位 (WDCS2 到 WDCS0)设置
3. 执行上述过程后，将 ACH 写入 WDTE，这样计数器被清零，允许重新计数。

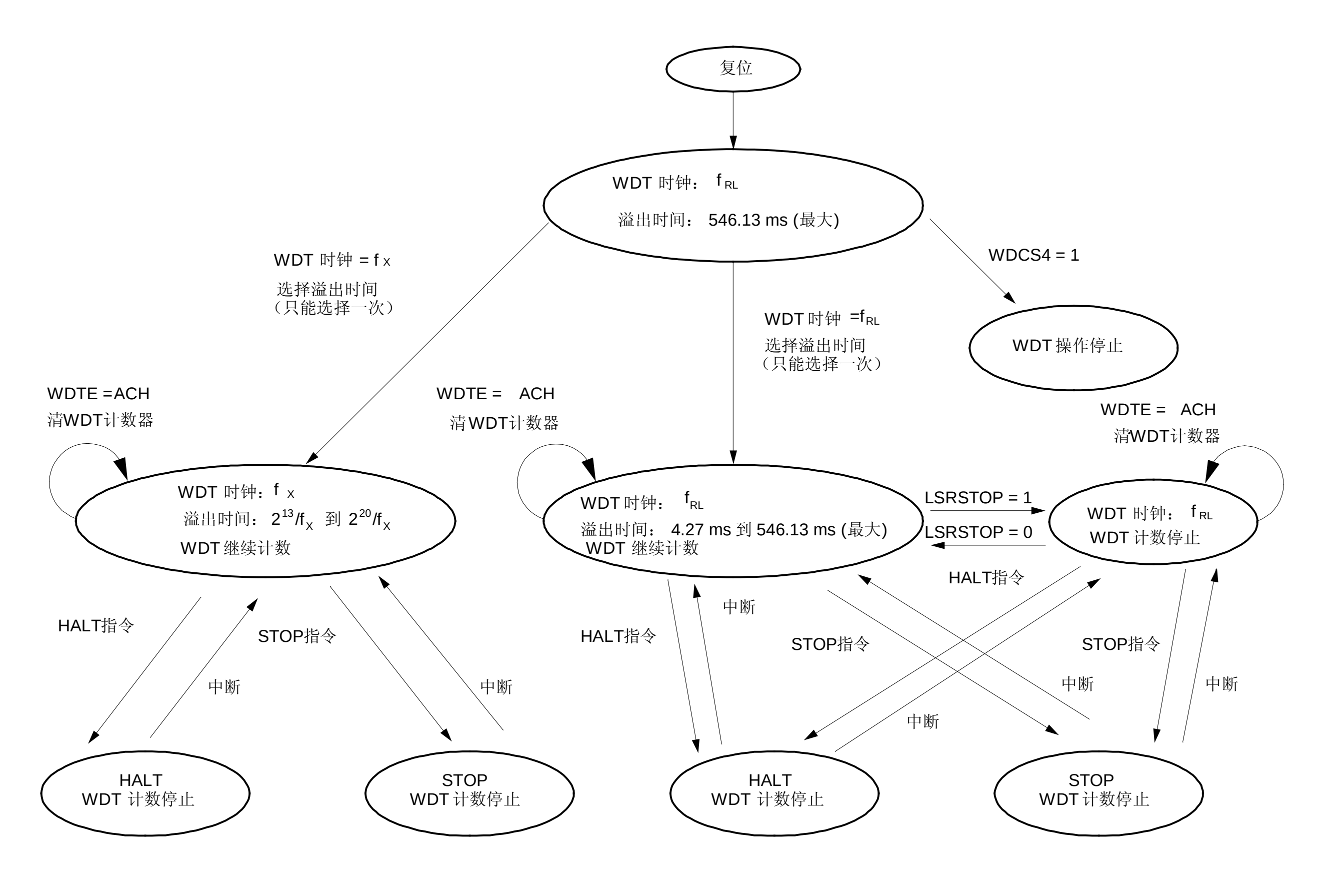
注
1. 只要对 WDTM 进行写入操作，看门狗定时器计数器就将被清零。
2. 第 7，6，5 位分别设置为 0，1，1，不能设置其他值。
3. 如果通过将 WDCS4 和 WDCS3 分别设置为 “1” 和 “x”来停止看门狗定时器操作，即使有下面的操作也不会产生内部复位。
   - 第二次写入 WDTM
   - 用 1 位存储器操作指令对 WDTE 进行操作
   - 给 WDTE 设置一个不同于“ACH”的值

注意事项　此模式中，在 HALT/STOP 指令执行期间看门狗定时器的操作被停止。 在释放 HALT/STOP 模式后，通过使用在 HALT/STOP 指令执行前 WDTM 设置的看门狗定时器操作时钟将使计数器再次启动。此时，计数器并没有被清零，而是保持原值。

要了解 STOP 和 HALT 模式下每一种状态下看门狗定时器的操作，可参阅 **9.4.3 STOP 模式下看门狗定时器的操作(当由可选字节选择为“可通过软件停止低速 Ring-OSC 时钟”时)** 和 **9.4.4 HALT 模式下看门狗定时器的操作(当由可选字节选择为“可通过软件停止低速 Ring-OSC 时钟”时)**。

参考如下所示状态转换图。

图 9-5. 当由可选字节选择“可通过软件停止低速 Ring-OSC 时钟”时的状态转换图



9.4.3 STOP 模式下看门狗定时器的操作（当由可选字节选择为 “可通过软件停止低速 Ring-OSC 时钟”）

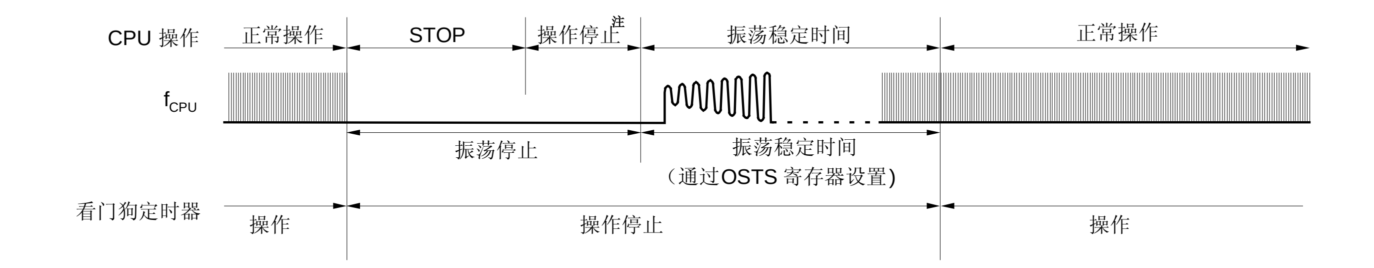
无论采用的是系统时钟还是低速 Ring-OSC 时钟，在执行 STOP 指令时看门狗定时器要停止计数。

(1) 当看门狗定时器操作时钟是外围硬件时钟(fx)，且执行 STOP 指令时。

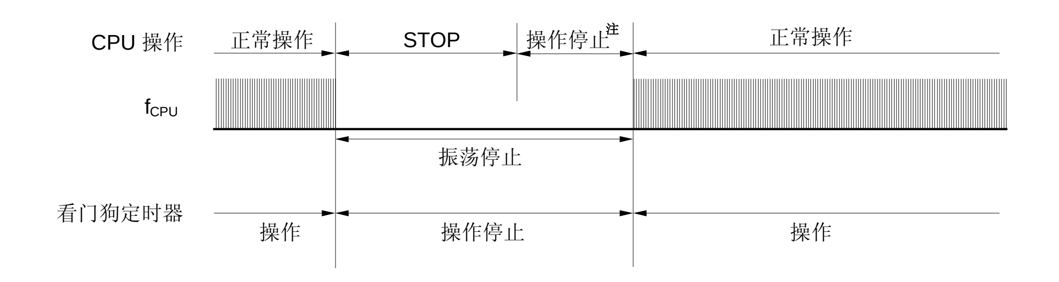
当执行 STOP 指令时，看门狗定时器的操作停止。在释放 STOP 模式后，操作停止 34 μs (典型值) (该值是在使用晶体/陶瓷振荡器时，由振荡器稳定时间选择寄存器（OSTS）设置的等待振荡器振荡稳定的时间)，然后在操作停止前使用操作时钟计数再次开始。此时，计数器没有被清零， 而是保持原值。

图 9-6. STOP 模式下的操作(WDT 操作时钟：外围硬件时钟)

〈1〉 CPU 时钟：晶体/陶瓷振荡器时钟



〈2〉 CPU 时钟：高速 Ring-OSC 时钟或外部时钟输入



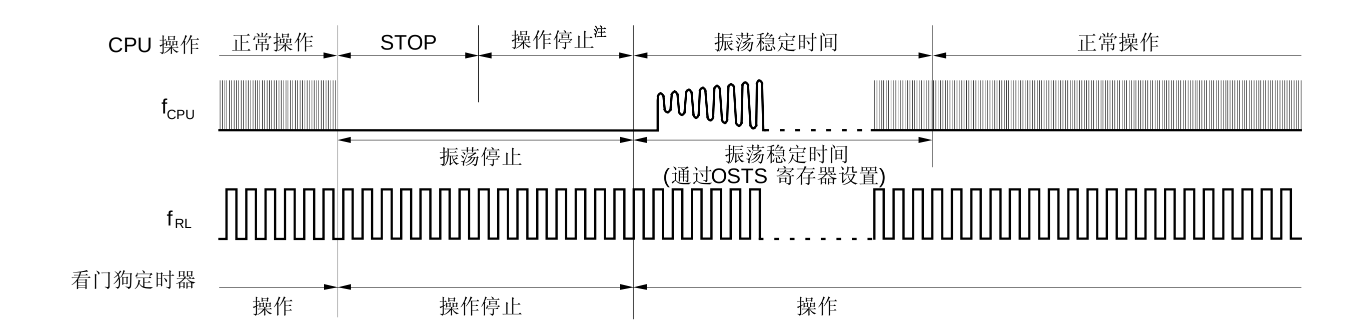
注 操作停止时间是 17 μs (最小值), 34 μs (典型值)和 67 μs (最大值)。

(2) 当看门狗定时器操作时钟是低速 Ring-OSC 时钟(f_{RL})，且执行 STOP 指令时。

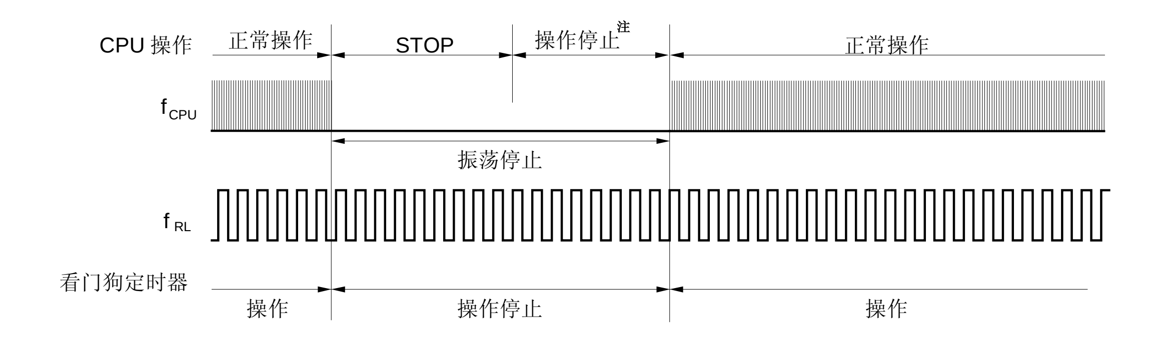
当执行 STOP 指令时，看门狗定时器的操作停止。在释放 STOP 模式后，操作停止 34 μs (TYP.)，然后使用操作停止前的操作时钟重新开始计数。此时，计数器没有被清零， 而是保持原值。

图 9-7.　STOP 模式下的操作(WDT 操作时钟：低速 Ring-OSC 时钟)

<1> CPU 时钟: 晶体/陶瓷振荡器时钟



<2> CPU 时钟: 高速 Ring-OSC 时钟或外部时钟输入

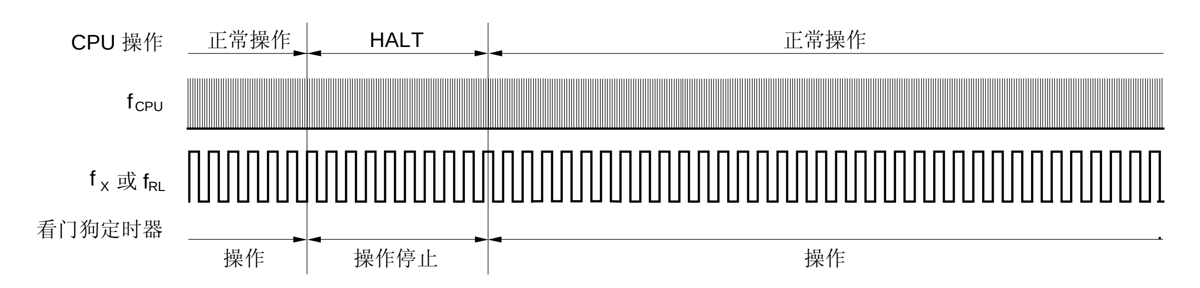


注　操作停止时间是 17 μs (最小值)，34 μs (典型值)和 67 μs (最大值)。

9.4.4　HALT 模式下的操作(当由可选字节选择为 “可通过软件停止低速 Ring-OSC 时钟”)

无论采用的是系统时钟(f_X)还是低速 Ring-OSC 时钟(f_{RL})，在执行 HALT 指令时看门狗定时器要停止计数。在释放 HALT 模式后，使用操作停止前的操作时钟重新开始计数。此时，计数器没有被清零， 而是保持原值。

图 9-8.　HALT 模式下的操作



第十章　A/D 转换器

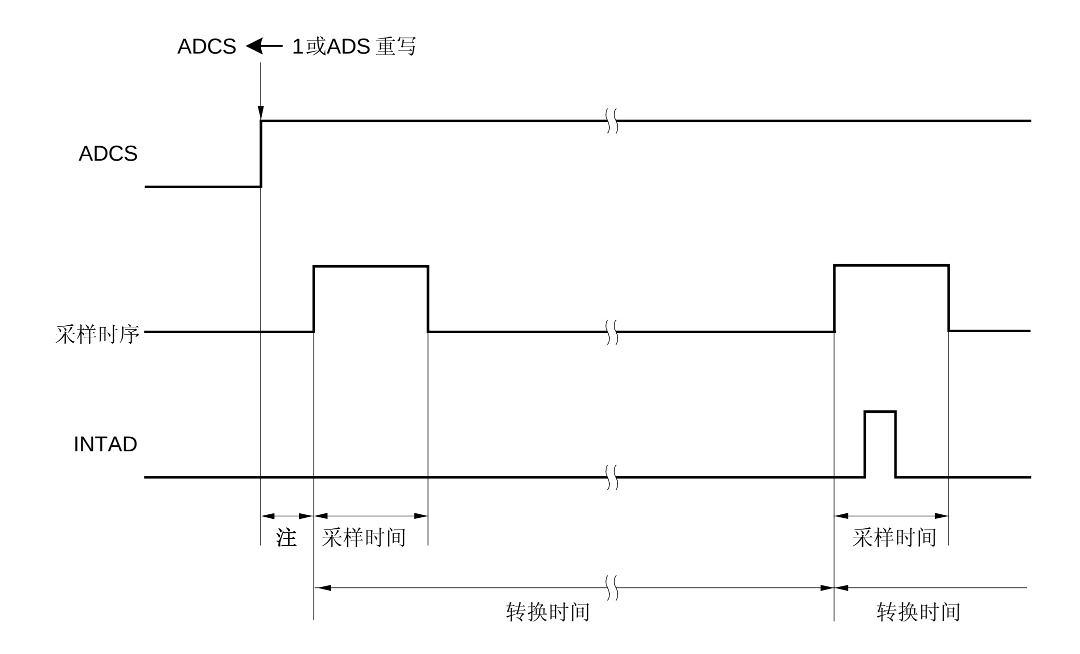
10.1 A/D 转换器的功能

A/D 转换器将模拟信号转换成数字信号，由 4 个通道(ANI0 ～ ANI3)组成，具有 10 位分辨率。
A/D 转换器有以下几个功能。

- 10 位分辨率 A/D 转换
  从模拟输入 ANI0～ANI3 中选择一个通道，重复执行 10 位分辨率 A/D 转换。每次 A/D 转换结束，都会产生一个中断请求(INTAD)。

图 10-1 为 A/D 采样和 A/D 转换的时序图，表 10-1 为采样时间和 A/D 转换的时间。

图 10-1.　A/D 转换器采样和 A/D 转换的时序



注　ADCS 上升到采样开始需要 2～3 个时钟。

表 10-1. 采样时间和 A/D 转换时间

FR2	FR1	FR0	参考电压范围[注 1]	采样时间[注 2]	转换时间[注 3]	f_{XP} = 8 MHz		f_{XP} = 10 MHz	
						采样时间[注 2]	转换时间[注 3]	采样时间[注 2]	转换时间[注 3]
0	0	0	AV_{REF} ≥ 4.5 V	12/f_{XP}	36/f_{XP}	1.5μs	4.5μs	1.2μs	3.6μs s
0	0	1	AV_{REF} ≥ 2.85 V	24/f_{XP}	48/f_{XP}	3.0μs	6.0μs	禁止设置 (2.4μs)	禁止设置 (4.8μs)
0	1	0	AV_{REF} ≥ 2.7 V	48/f_{XP}	72/f_{XP}	禁止设置 (6.0μs)	禁止设置 (9.0μs)	禁止设置 (4.8μs)	禁止设置 (7.2μs)
0	1	1	AV_{REF} ≥ 2.7 V	88/f_{XP}	112/f_{XP}	11.0μs	14.0μs	禁止设置 (8.8μs)	禁止设置 (11.2μs)
1	0	0	AV_{REF} ≥ 4.5 V	24/f_{XP}	72/f_{XP}	3.0μs	9.0μs	2.4μs	7.2μs
1	0	1	AV_{REF} ≥ 2.85 V	48/f_{XP}	96/f_{XP}	6.0μs	12.0μs	4.8μs	9.6μs
1	1	0	AV_{REF} ≥ 2.7 V	96/f_{XP}	144/f_{XP}	12.0μs	18.0μs	禁止设置 (9.6μs)	禁止设置 (14.4μs)
1	1	1	AV_{REF} ≥ 2.7 V	176/f_{XP}	224/f_{XP}	22.0μs	28.0μs	17.2μs	22.4μs

注 1. 设置 FR2, FR1 和 FR0 值时，要与参考电压范围一致，并且满足以下注 2 和注 3 的条件。

例如 当 AV_{REF} ≥ 2.7Ｖ 时

- 设置 FR2，FR1 和 FR0 = 0，1，1 或 1，1，1。
- 采样时间大于或等于 11.0μs，A/D 转换时间大于或等于 14.0μs 并小于等于 100μs。

2. 设置采样时间：

- AV_{REF} ≥ 4.5 V：⩾1.0μs
- AV_{REF} ≥ 4.0 V：⩾2.4μs
- AV_{REF} ≥ 2.85 V：⩾3.0μs
- AV_{REF} ≥ 2.7 V：⩾11.0μs

3. 设置 A/D 转换时间：

- AV_{REF} ≥ 4.5 V：⩾3.0μs 且<100μs
- AV_{REF} ≥ 4.0 V：⩾4.8μs 且<100μs
- AV_{REF} ≥ 2.85 V：⩾6.0μs 且<100μs
- AV_{REF} ≥ 2.7 V：⩾14.0μs 且<100μs

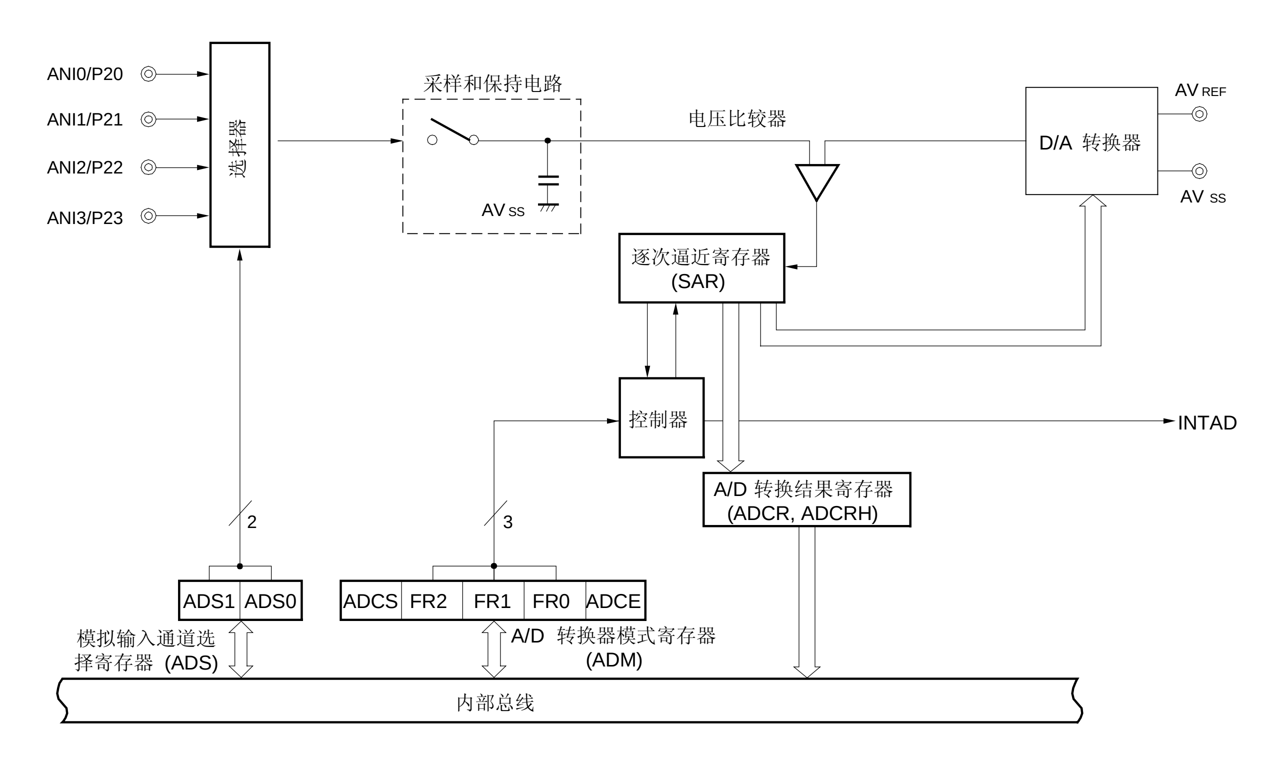
注意事项 以上采样时间和转换时间没有包括时钟频率误差。选择采样时间和转换时间时要将时钟频率误差考虑在内。

备注 1. f_{XP}： 外部硬件时钟的振荡频率

2. 转换时间包括采样时间和与采样值进行连续地比较，直到转换结果输出的时间。

图 10-2 为 A/D 转换器的框图。

图 10-2. A/D 转换器的框图



10.2 A/D 转换器配置

A/D 转换器由以下硬件组成。

表 10-2. A/D 转换器中使用的寄存器

项目	配置
寄存器	10 位 A/D 转换结果寄存器(ADCR) 8 位 A/D 转换结果寄存器(ADCRH) A/D 转换模式寄存器(ADM) 模拟输入通道选择寄存器(ADS) 端口模式控制寄存器 2 (PMC2) 端口模式寄存器 2 (PM2)

(1) ANI0 至 ANI3 引脚

ANI0 至 ANI3 是 4 通道 A/D 转换器的模拟输入引脚。它们输入模拟信号，转换为数字信号。除了那些被模拟输入通道寄存器 (ADS) 选为模拟输入的引脚外，其他引脚均可用作输入端口引脚。

(2) 采样 & 保持电路

在 A/D 转换开始后，采样&保持电路对由选择器选择的模拟输入引脚的输入信号进行采样，并在 A/D 转换期间保持采样到的模拟输入电压值。

(3) D/A 转换器

D/A 转换器用于连接 AV_{REF} 和 AV_{SS}，并产生一个电压，与模拟输入信号比较。

(4) 电压比较器

电压比较器用于比较采样模拟输入电压和 D/A 转换器的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器用于比较采样的模拟电压和 D/A 转换器的电压，并从最高有效位(MSB)开始得到转换结果。

当电压值转换为数字值直到最低有效位 (LSB) (A/D 转换结束)时，将 SAR 内容传送给 A/D 转换结果寄存器 (ADCR)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束时，将 A/D 转换的结果从逐次逼近寄存器传送给该寄存器中，并将转换结果保存在 ADCR 的低 10 位中 (高 6 位固定为 0)。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束时，将 A/D 转换的结果从逐次逼近寄存器传送给该寄存器中，并将转换结果保存在 ADCR 的高 8 位中

(8) 控制器

A/D 转换结束，产生中断 INTAD。

(9) AV_{REF} 引脚

该引脚给 A/D 转换器输入模拟电源/参考电压。当不使用 A/D 转换器时，直接与 V_{DD} 引脚相连。

ANI0 至 ANI3 引脚的输入信号基于 AV_{REF} 和 AV_{SS} 之间的电压被转换成数字信号。

(10) AV_{SS} 引脚

该引脚为地引脚。无论使用 A/D 转换器与否，一直保持该引脚与 V_{SS} 引脚同电位。

(11) A/D 转换模式寄存器 (ADM)

此寄存器用来设置转换模拟输入信号的转换时间，和开始或停止转换操作。

(12) 模拟输入通道选择寄存器 (ADS)

此寄存器用来选择将模拟输入电压转换为数字信号的输入端口。

(13) 端口模式控制寄存器 2 (PMC2)

当 P20/ANI0 至 P23/ANI3 引脚作为 A/D 转换器的模拟输入引脚时 ，使用该寄存器。

10.3 A/D 转换器中使用的寄存器

A/D 转换器使用以下六种寄存器：

- A/D 转换模式寄存器(ADM)
- 模拟输入通道选择寄存器(ADS)
- 10 位 A/D 转换结果寄存器(ADCR)
- 8 位 A/D 转换结果寄存器(ADCRH)
- 端口模式控制寄存器 2 (PMC2)
- 端口模式寄存器 2 (PM2)

(1) A/D 转换器模式寄存器(ADM)

此寄存器用来设置模拟输入的转换时间，和开始/停止转换操作。

可通过 1 位或 8 位存储器操作指令设置 ADM。

复位后该寄存器清零（00H）。

图 10-3.　A/D 转换器模式寄存器 (ADM) 的格式

地址： FF80H　　复位后： 00H　　R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D 转换操作控制
0	停止转换操作
1 [注1]	开始转换操作

FR2	FR1	FR0	参考电压范围[注2]	采样时间[注3]	转换时间[注4]	f_{XP} = 8 MHz		f_{XP} = 10 MHz	
						采样时间[注3]	转换时间[注4]	采样时间[注3]	转换时间[注4]
0	0	0	AV_{REF} ≥ 4.5 V	12/f_{XP}	36/f_{XP}	1.5μs	4.5μs	1.2μs	3.6μs
0	0	1	AV_{REF} ≥ 2.85 V	24/f_{XP}	48/f_{XP}	3.0μs	6.0μs	禁止设置 (2.4μs)	禁止设置 (4.8μs)
0	1	0	AV_{REF} ≥ 2.7 V	48/f_{XP}	72/f_{XP}	禁止设置 (6.0μs)	禁止设置 (9.0μs)	禁止设置 (4.8μs)	禁止设置 (7.2μs)
0	1	1	AV_{REF} ≥ 2.7 V	88/f_{XP}	112/f_{XP}	11.0μs	14.0μs	禁止设置 (8.8μs)	禁止设置 (11.2μs)
1	0	0	AV_{REF} ≥ 4.5 V	24/f_{XP}	72/f_{XP}	3.0μs	9.0μs	2.4μs	7.2μs
1	0	1	AV_{REF} ≥ 2.85 V	48/f_{XP}	96/f_{XP}	6.0μs	12.0μs	4.8μs	9.6μs
1	1	0	AV_{REF} ≥ 2.7 V	96/f_{XP}	144/f_{XP}	12.0μs	18.0μs	禁止设置 (9.6μs)	禁止设置 (14.4μs)
1	1	1	AV_{REF} ≥ 2.7 V	176/f_{XP}	224/f_{XP}	22.0μs	28.0μs	17.6μs	22.4μs

ADCE	比较操作控制[注5]
0 [注1]	停止比较器操作
1	使能比较器操作

备注　1. f_{XP}: 外部硬件时钟的振荡频率

2. 转换时间包括采样时间和对采样值进行成功地比较，直到转换结果输出的时间。

注　1. 即使当 ADCE = 0 (比较器操作停止)时，如果 ADCS =1， A/D 转换操作也会开始。但是，第一次转换的数据不在保证的范围之内，因而应该忽略它。

2. 设置 FR2, FR1 和 FR0 值时，要与参考电压范围一致，并满足以下 **注 3** 和 **注 4** 的条件。

例如　　当 AV_{REF} ≥ 2.7 V 时

- 设置 FR2，FR1 和 FR0 = 0，1，1 或 1，1，1。
- 采样时间大于或等于 11.0 μs，A/D 转换时间大于或等于 14.0 μs 且小于等于 100 μs。

注　3. 设置采样时间：
- AVREF ≥ 4.5 V： ⩾1.0μs
- AVREF ≥ 4.0 V： ⩾2.4μs
- AVREF ≥ 2.85 V：⩾3.0μs
- AVREF ≥ 2.7 V： ⩾11.0μs

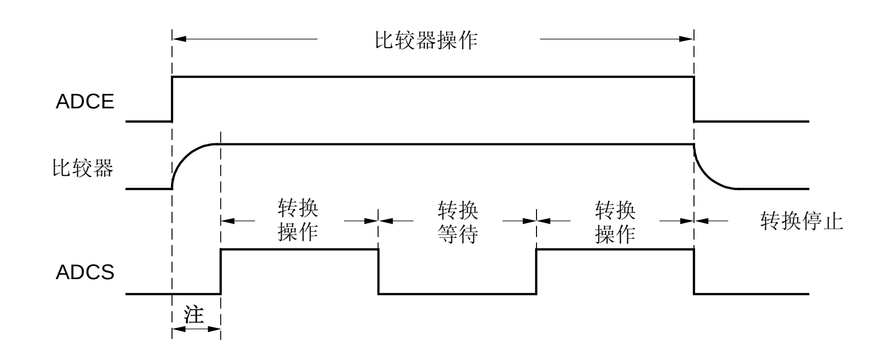
4. 设置 A/D 转换时间：
- AVREF ≥ 4.5 V： ⩾3.0μs 且<100μs
- AVREF ≥ 4.0 V： ⩾4.8μs 且<100μs
- AVREF ≥ 2.85 V：⩾6.0μs 且<100μs
- AVREF ≥ 2.7 V： ⩾14.0μs 且<100μs

5. 比较操作通过 ADCS 和 ADCE 来控制，从开始操作到操作稳定需要 1μs 的时间。因此，从 ADCE=1，经过 1μs 或更长时间，再将 ADCS 设置为 1，此时的转换结果比没有等待的转换结果要好。如果 ADCS 设置为 1 前没有等待 1 μs 或更长时间，就应该忽略第一次转换的数据。

表 10-3.　ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流电源消耗)
0	1	转换等待模式 (仅有比较器电源消耗)
1	×	转换模式

图 10-4.　当使用比较器时的时序图



注　从 ADCE 上升沿到 ADCS 的上升沿至少要等待 1μs 的时间，内部电路才能稳定。

注意事项 1. 以上采样时间和转换时间没有包括时钟频率误差。选择转换时间时还要将时钟频率误差考虑在内。

2. 当 A/D 转换结束(ADCS = 0)时，对 ADM 中除了 ADCS 的位进行操作后，要执行两条 NOP 指令或一条相当于 2 个机器周期的指令，再设置 ADCS=1，A/D 转换重新开始。

3. 在重写 FR0 至 FR2 时，A/D 转换必须停止 (ADCS = 0)。

4. 一定要把第 6 位、第 2 位和第 1 位清 0。

(2) 模拟输入通道选择寄存器(ADS)

此寄存器选择被转换的模拟电压输入端口。

可通过 1 位或 8 位存储器操作指令设置 ADS。

复位后该寄存器清零（00H）。

图 10-5.　模拟输入通道选择寄存器 (ADS)的格式

地址: FF81H　复位后: 00H　读/写

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	0	ADS1	ADS0

ADS1	ADS0	模拟输入通道选择寄存器
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	ANI3

注意事项　ADS 的第 2 位到第 7 位必须清 0。

(3) 10 位 A/D 转换结果寄存器(ADCR)

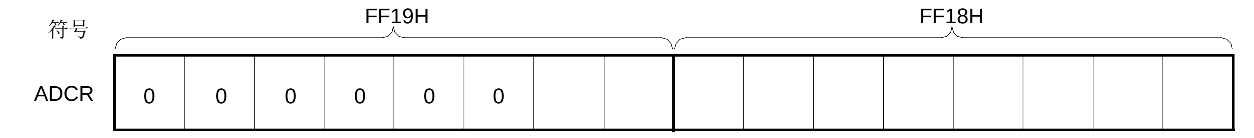
此寄存器是一个 16 位的寄存器，用来存储 A/D 转换的结果。该寄存器的高 6 位固定为 0。每次 A/D 转换结束后，转换的结果从逐次比较寄存器中读出存储到 ADCR 中，存储从 FF19H 的位 1 开始，FF19H 存放转换结果的高 2 位，FF18H 存放转换结果的低 8 位。

ADCR 的值可以通过 16 位存储器操作指令读取。

复位后 ADCR 的值不确定。

图 10-6.　10 位 A/D 转换结果寄存器 (ADCR)的格式

地址: FF18H, FF19H　复位后: 不确定　只读



注意事项　当对 A/D 转换模式寄存器（ADM）和模拟输入通道选择寄存器（ADS）进行写操作时，ADCR 的内容可能不确定。在对 ADM 和 ADS 执行写操作之前，要在转换完成后读取转换结果。在其他时间读取可能读到错误的转换结果。

(4) 8 位 A/D 转换结果寄存器(ADCRH)

此寄存器是一个 8 位的寄存器，用来存储 A/D 转换的结果。它存储 10 位转换结果的高 8 位。

可通过 8 位存储器操作指令读取 ADCRH。

复位后 ADCRH 的值不确定。

图 10-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

地址：FF1AH 复位后: 不确定 只读

符号	7	6	5	4	3	2	1	0
ADCRH								

(5) 端口模式控制寄存器 2 (PMC2) 和端口模式寄存器 2 (PM2)

当使用 P20/ANI0 至 P23/ANI3 引脚作为模拟输入时，设置 PMC20 至 PMC23 和 PM20 至 PM23 为 1。此时，P20 至 P23 的输出缓存可以为 0 或 1。

PM2 和 PMC2 可通过 1 位或 8 位存储器操作指令设置。

复位后 PMC2 为 00H，PM2 为 FFH。

图 10-8. 端口模式控制寄存器 2 (PMC2)的格式

地址： FF84H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	0
PMC2	0	0	0	0	PMC23	PMC22	PMC21	PMC20

PMC2n	操作模式选择 (n = 0 ～ 3)
0	端口模式
1	A/D 转换器模式

图 10-9. 端口模式寄存器 2 (PM2)的格式

地址：FF22H 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	Pmn 引脚 I/O 模式选择 (n = 0～ 3)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

注意事项 当 PMC20 至 PMC23 被设置为 1 时，P20/ANI0 至 P23/ANI3 引脚不能被用作端口引脚。

10.4 A/D 转换器的操作

10.4.1 A/D 转换器的基本操作

〈1〉 使用模拟输入通道选择寄存器（ADS）为 A/D 转换选择一个通道。
〈2〉 设置 ADCE 为 1 并等待 1 μs 或更长。
〈3〉 执行两条 NOP 指令或一条相当于两个机器周期的指令。
〈4〉 设置 ADCS 为 1 并开始转换操作。
(〈5〉至〈11〉的操作由硬件实现)
〈5〉 采样&保持电路对选中的模拟输入通道的输入电压进行采样。
〈6〉 经过一段时间的采样后，采样&保持电路处于保持状态，在 A/D 转换结束前，一直保持输入的模拟电压。
〈7〉 设置逐次比较寄存器的第 9 位。通过分接选择器将 D/A 转换电压置为（1/2）AV_{REF}。
〈8〉 电压比较器比较 D/A 转换电压和模拟输入电压。如果模拟输入电压大于（1/2）AV_{REF}，SAR 中 MSB 为 1。如果模拟输入电压小于（1/2）AV_{REF}，SAR 中 MSB 为 0。
〈9〉 接下来，SAR 的第 8 位自动置 1，并进入下一个比较过程。根据第 9 位的预置值选择 D/A 转换电压。具体描述如下：

- 第 9 位= 1： (3/4) AV_{REF}
- 第 9 位= 0： (1/4) AV_{REF}

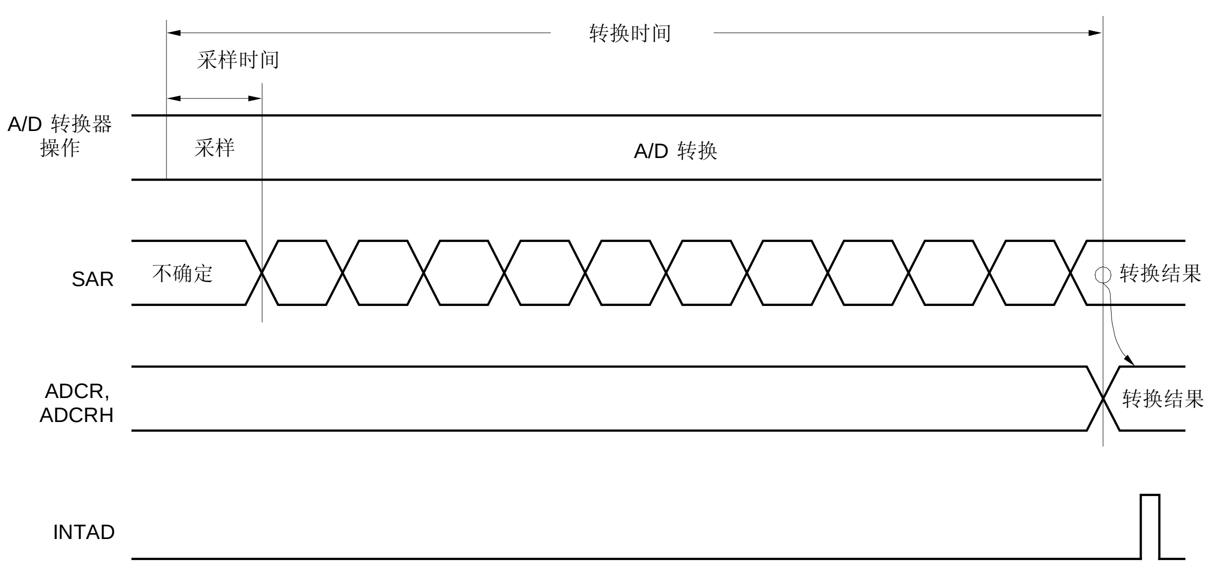
比较分接电压和模拟输入电压，并设置 SAR 的第 8 位，如下所示。

- 模拟输入电压≥分接电压：　第 8 位为 1
- 模拟输入电压<分接电压：　第 8 位为 0

〈10〉 按此方式继续比较，直至 SAR 的第 0 位。
〈11〉 全部 10 位都比较完成后，有效的数字结果保存在 SAR 中，并将该结果传输给 A/D 转换结果寄存器（ADCR，ADCRH）中，然后锁存。
同时，产生 A/D 转换结束中断请求（INTAD）。
〈12〉 重复执行〈5〉 到 〈11〉，直到 ADCS 清 0。
ADCS 清零， A/D 转换停止。
如要从 ADCE=1 的状态重新开始 A/D 转换，从步骤〈3〉开始。如要从 ADCE=0 的状态重新开始 A/D 转换，从步骤〈2〉开始。

备注　可使用以下两种类型的 A/D 转换结果寄存器。
〈1〉 ADCR (16 位)：　存储 10 位 A/D 转换值。
〈2〉 ADCRH (8 位)：　存储 8 位 A/D 转换值。

图 10-10.　A/D 转换器的基本操作



在用软件将 A/D 转换器模式寄存器（ADM）的第 7 位(ADCS)设置为 0 之前，连续执行 A/D 转换操作。

如果在 A/D 转换操作期间，对 ADM 或模拟输入通道选择寄存器（ADS）执行写操作，则转换操作被初始化。并且若 ADCS 置 1，转换将再次从头开始。

产生复位信号后，A/D 转换结果寄存器（ADCR，ADCRH）的值不确定。

10.4.2 输入电压和转换结果

输入到模拟输入引脚（ANI0～ANI3）的模拟输入电压和理论 A/D 转换结果（存储在 10 位 A/D 转换结果寄存器（ADCR））的关系如下。

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

或

$$(ADCR - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{1024}$$

这里，INT( ) ：该函数返回括号中数值的整数部分

V_{AIN}：　模拟输入电压

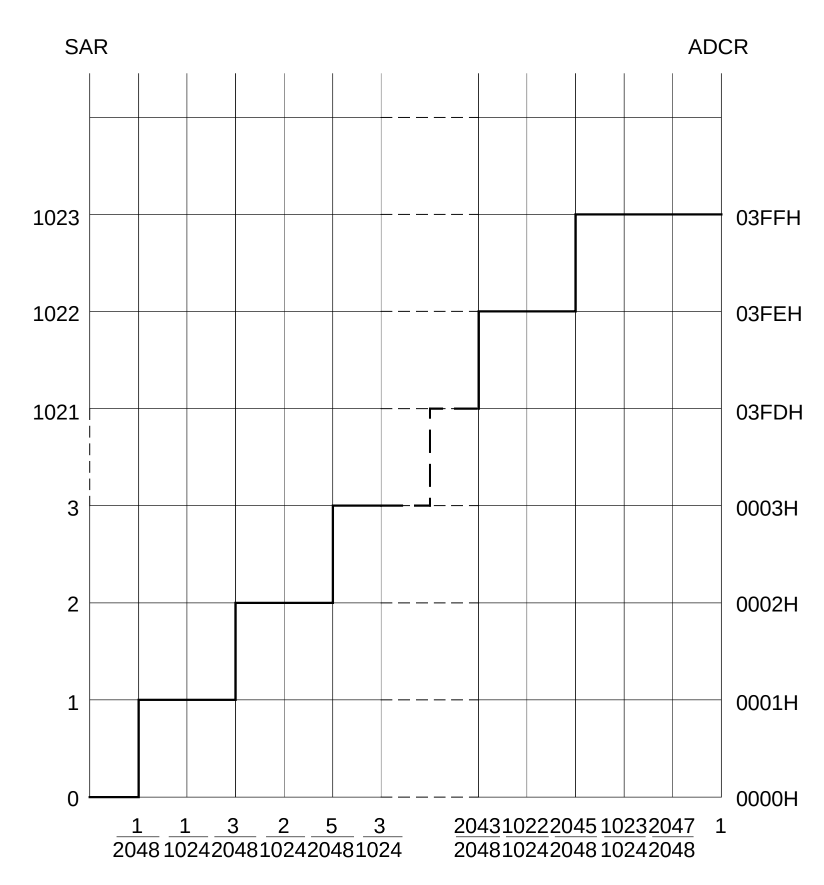
AV_{REF}：　AV_{REF} 引脚电压

ADCR：　10 位 A/D 转换结果寄存器 (ADCR) 的值

SAR：　逐次逼近寄存器

图 10-11 显示模拟输入电压和 A/D 转换结果之间的关系。

图 10-11. 模拟输入电压和 A/D 转换结果之间的关系



10.4.3 A/D 转换操作模式

由模拟输入通道选择寄存器（ADS），从 ANI0 至 ANI3 中选择一个模拟输入通道，并执行 A/D 转换。

(1) A/D 转换操作

通过设置 A/D 转换模式寄存器 ADM 的第 7 位（ADCS）为 1，可启动对电压的 A/D 转换操作。该电压由模拟输入通道选择寄存器（ADS）选择的模拟输入引脚输入。

当 A/D 转换完成时，转换结果存储在 A/D 转换结果寄存器（ADCR，ADCRH），并且产生一个中断请求信号（INTAD）。如果 A/D 转换已经开始而一次 A/D 转换结束时， 下一次 A/D 转换立即开始。A/D 转换操作一直持续到有新数据写入到 ADS 为止。

若在 A/D 转换期间，ADM 或 ADS 值被重写，则要停止当前的 A/D 转换操作，重新开始。

若在 A/D 转换期间，ADCS 被置为 0，则 A/D 转换立即停止。此时的转换结果不确定。

图 10-12. A/D 转换操作

备注 1. n = 0 ～ 3
2. m = 0 ～ 3

设置方式描述如下。

〈1〉 设 A/D 转换模式寄存器（ADM）的第 0 位(ADCE)为 1。
〈2〉 通过模拟输入通道选择寄存器（ADS）的第 1 位和第 0 位（ADS1，ADS0）和 ADM 的第 5 位到第 3 位（FR2～FR0），来选择通道和转换时间。
〈3〉 执行两条 NOP 指令或一条相当于 2 个机器周期的指令。
〈4〉 设置 ADM 的第 7 位(ADCS)为 1，来启动 A/D 转换。
〈5〉 产生中断请求信号(INTAD)。
〈6〉 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR, ADCRH)。

〈改变通道〉

〈7〉 使用 ADS 的第 1 位和第 0 位(ADS1, ADS0)改变通道。
〈8〉 产生中断请求信号(INTAD) 。
〈9〉 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR, ADCRH)。

〈 A/D 转换完成〉

〈10〉 ADCS 清 0。
〈11〉 ADCE 清 0。

注意事项 1. 必须确保步骤〈1〉 到 〈4〉 的操作时间为 1 μs 或更长。
2. 若将步骤〈1〉和〈2〉次序颠倒，也不会有问题。
3. 〈1〉 可以被忽略。 但在这种情况下中当步骤〈4〉执行后忽略第一个转换结果。
4. 步骤〈5〉到〈8〉所经历的时间与 ADM 的第 5 位到第 3 位(FR2～FR0)设置的转换时间不同。步骤〈7〉到〈8〉所经历的时间即为转换时间，由 FR2 到 FR0 设置。

10.5 A/D 转换器特性表的阅读方法

以下，介绍 A/D 转换器的专用术语

(1) 分辨率

这是可识别的最小模拟输入电压。每位数字输出的模拟输入电压的百分比，叫 1LSB（最低有效位）。满度的 1 LSB 的百分比用 %FSR（满度范围）表示。

当分辨率为 10 位时 1LSB 表示如下。

$$1LSB = 1/2^{10} = 1/1024$$
$$= 0.098\%FSR$$

准确度和分辨率无关，而由总误差决定。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差值。
零度误差、满度误差、积分线性误差和微分线性误差等组合起来表示总误差。
注意量化误差不属于特性表中总误差的范围。

(3) 量化误差

当模拟值转换为数字值时，通常会产生 a ±1/2LSB 的误差。 在一个 A/D 转换器中，相差±1/2LSB 的模拟输入电压被转换成相同数字代码，因此量化误差不可避免。
注意量化误差不属于特性表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 10-13.　总误差

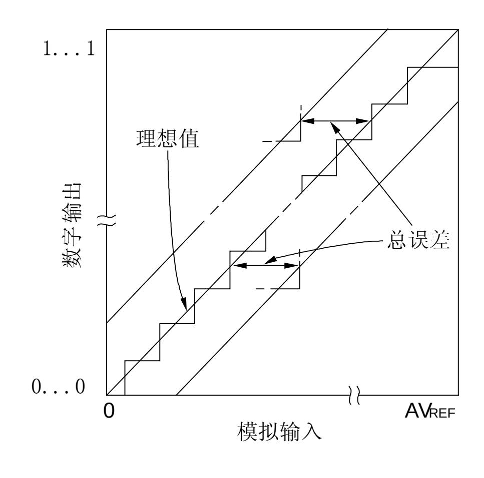
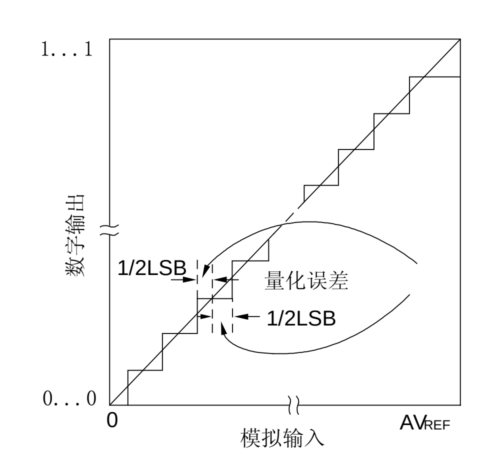


图 10-14.　总误差



(4) 零度误差

零度误差表示当数字输出范围在 0......000 ～ 0......001 之间时，模拟输入电压的实际测量值和理论值(1/2LSB)之间的误差。
如果实际测量值大于理论值，它表示为当数字输出范围在 0……001 ～ 0……010 之间时，模拟输入电压的实际测量值和理论值(3/2LSB)之间的误差。

(5) 满度误差

满度误差表示当数字输出范围在 1......110 ～ 1......111 时，模拟输入电压的实际测量值和理论值(满度 – 3/2LSB)之间的误差。

(6) 积分线性误差

积分线性误差说明了转换特征偏离理论线性关系的程度。它表示当零度误差和满度误差为 0 时，实际测量值和理论线性之间误差的最大值。

(7) 微分线性误差

当代码输出的理论宽度为 1LSB 时，微分线性误差表示实际测量值和理论值之间的误差。

图 10-15. 零度误差

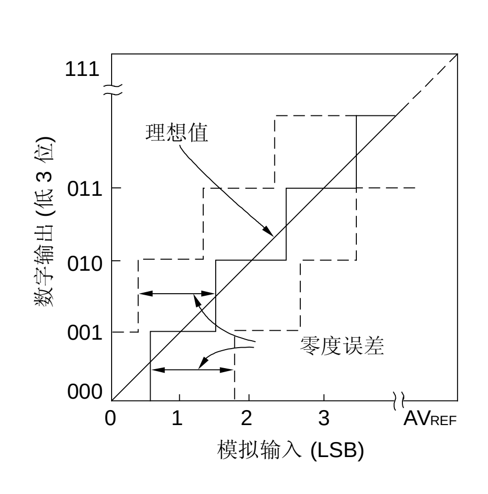


图 10-16. 满度误差

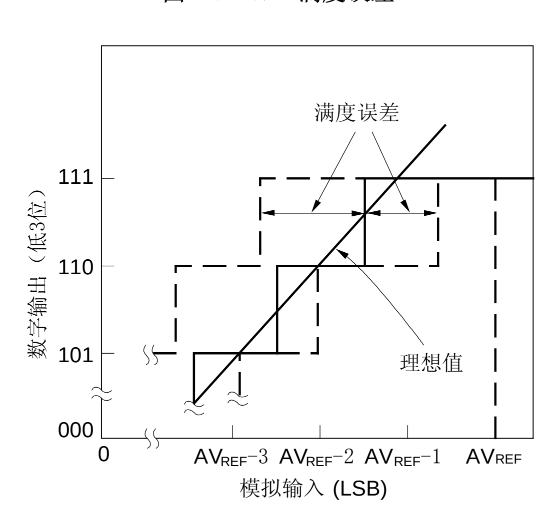


图 10-17. 积分线性误差

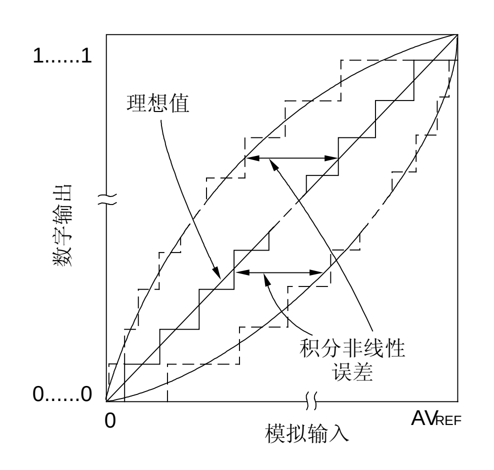
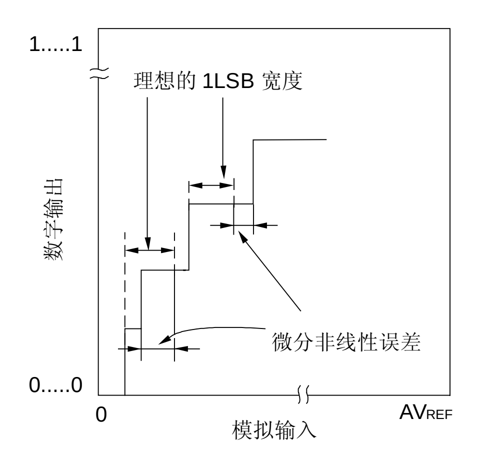


图 10-18. 微分线性误差



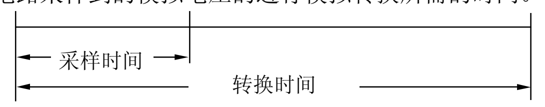
(8) 转换时间

转换时间表示开始采样到获取数字输出所经历的时间。

在特性表中采样时间包含在转换时间中。

(9) 采样时间

采样时间表示通过采样&保持电路采样到的模拟电压的进行模拟转换所需的时间。



10.6 A/D 转换器使用注意事项

(1) STOP 模式下的工作电流

在 STOP 模式下，A/D 转换停止操作。此时，将 A/D 转换模式寄存器（ADM）的位 7（ADCS）和位 0（ADCE）清 0，可以降低操作电流。

(2) ANI0 至 ANI3 的输入范围

观察 ANI0 至 ANI3 的输入电压的额定范围。如果输入到模拟输入通道的电压为 V_{DD} 或更高，或者为 V_{SS} 或更低（即使在额定的最大范围内），则该通道的转换值是不确定的。此外，其他通道的转换值也可能受影响。

(3) 冲突操作

〈1〉 转换结束后，指令对 A/D 转换结果寄存器（ADCR，ADCRH）的写操作和 ADCR，ADCRH 的读操作同时进行，就为冲突操作。
ADCR,ADCRH 读操作优先。在读操作执行完后，新的转换结果将写入到 ADCR、ADCRH。

〈2〉 转换结束后，对 ADCR,ADCRH 写操作和 A/D 转换模式寄存器(ADM)写操作，或者对模拟输入通道选择寄存器(ADS)写操作同时进行，就为冲突操作。
ADM 或 ADS 写操作优先。如果 ADCR,ADCRH 的写操作不完成，就不会产生转换结束中断信号（INTAD）。

(4) 噪声解决办法

为了保持 10 位分辨率，必须注意输入到 V_{DD} 引脚和 ANI0 至 ANI3 引脚的噪声。

〈1〉 在低阻抗和高频反馈的电源上连接一个电容。

〈2〉 由于噪声的影响和模拟输入源的输出阻抗成正比，因此建议外接一个电容，如图 9-19 所示，以降低噪声。

〈3〉 在转换期间，不要将引脚 ANI0 到 ANI3 的 A/D 转换功能切换到它们的复用功能。

〈4〉 在转换开始后，立即设置 HALT 模式，可以提高转换精度。

图 10-19. 模拟输入引脚连接

(5) ANI0/P20 至 ANI3/P23

<1> 模拟输入引脚(ANI0 至 ANI3) 也可用作输入端口引脚 (P20 至 P23)。
当选择 ANI0 至 ANI3 进行 A/D 转换时，在转换过程中不要访问 P20 至 P23，否则会降低转换效果。

<2> 如果正在进行 A/D 转换的引脚相邻引脚有数字脉冲，则由于噪声耦合，有可能得不到预期的 A/D 转换值。因此，在进行 A/D 转换时不要在相邻引脚输入脉冲信号。

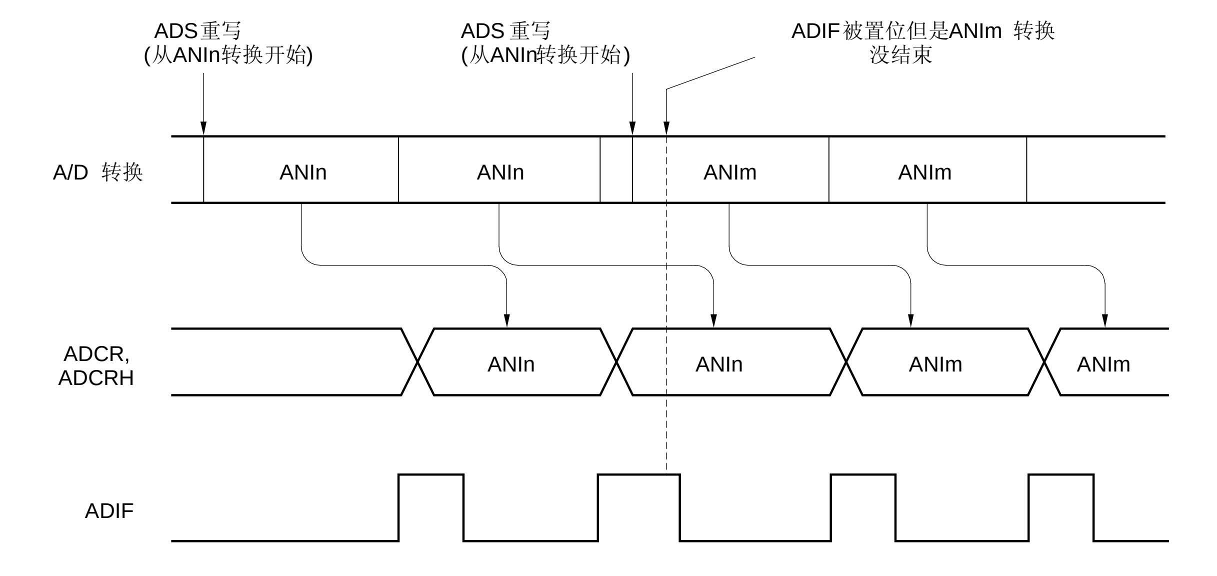
(6) ANI0 至 ANI3 引脚的输入阻抗

在 A/D 转换器中，采样期间内内部采样电容充电，采样时间接近转换时间的 1/6。
由于在采样期间有漏电流和电容充电的电流，因此输入阻抗在采样期间或其他时候都有波动。
如果在使用参考电压的最短转换时间，要进行充足的采样。建议模拟输入源的输出阻抗为 1kΩ或更小，或者在引脚连接 0.01μF 至 0.1μF 的电容(见 图 10-19)。

(7) 中断请求信号 (ADIF)

即使模拟输入通道选择寄存器（ADS）的值被修改，中断请求信号 (ADIF)也不能清零。
因此，如果在 A/D 转换期间有模拟输入引脚改变，那么在重写 ADS 前，A/D 转换的结果和模拟输入改变前的 ADIF 可能被设置。注意在 ADS 重写后立即读取 ADIF 时，尽管模拟输入改变后的 A/D 转换并未结束，也会设置 ADIF。
在 A/D 转换结束要重新开始转换时，要在重新开始转换前先将 ADIF 清 0。

图 10-20. A/D 转换结束中断请求产生的时序



备注 1. n = 0 ～ 3
2. m = 0 ～ 3

(8) A/D 转换刚开始的转换结果

A/D 转换开始后，如果在 ADCE 置 1 后 1 μs 内对 ADCS 置 1，或在 ADCE 为零时对 ADCS 置 1，那么第一次 A/D 转换的值可能不在额定范围内。可采取措施，如将 A/D 转换结束中断请求（INTAD）挂起，并删除第 1 次转换结果。

(9) A/D 转换结果寄存器 (ADCR, ADCRH) 的读取操作

当对 A/D 转换器模式寄存器（ADM）和模拟输入通道选择寄存器（ADS）进行读取时，ADCR 和 ADCRH 的内容可能变得不确定。要在 ADM 和 ADS 进行写操作前并且转换结束后，读取转换结果。若采用和上述不同的时序，可能会读取到不正确的转换结果。

(10) 内部等效电路

模拟输入模块的等效电路如下所示。

图 10-21.　ANIn 引脚的内部等效电路

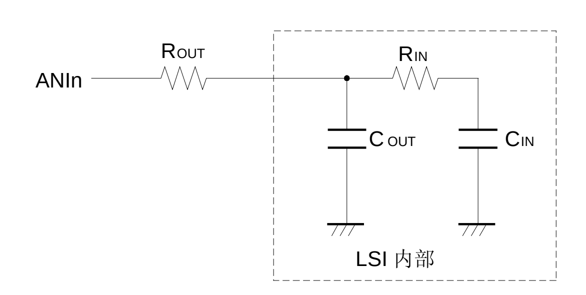


表 10-4.　等效电路的电阻和电容（参考值）

AVREF	ROUT	RIN	COUT	CIN
4.5 V ≤ AVREF ≤ 5.5 V	1 kΩ	3 kΩ	8 pF	15 pF
2.7 V ≤ AVREF < 4.5 V	1 kΩ	60 kΩ	8 pF	15 pF

备注　1. 表 10-4 所显示的电阻和电容值仅为参考值

2. n = 0 ～ 3

3. ROUT: 信号源阻抗

RIN:　模拟输入等效电阻

COUT:　内部引脚电容

CIN:　模拟输入等效电容

11.1 串行接口 UART6 的功能

串行接口 UART6 有以下两种模式。

(1) **操作停止模式**

在不进行串行通信时可采用此模式，以降低功率消耗。

需要了解详细信息，可参见 11.4.1 **操作停止模式**。

(2) **异步串行接口(UART)模式**

该模式支持 LIN (局部互联网)-总线。该模式的功能概括如下。

需要了解详细信息，可参见 11.4.2 **异步串行接口 (UART) 模式** 和 11.4.3 **专用波特率发生器**。

- 双引脚配置　　TxD6: 发送数据输出引脚
  RxD6: 接收数据输入引脚
- 可选择通信数据的宽度为 7 ～ 8 位。
- 专用内置 8 位 波特率发生器允许设置任意大小的波特率。
- 可分别执行发送和接收操作。
- 可选 MSB-first 或 LSB-first 通信。
- 反向传送操作。
- 13 ～ 20 位为同步中断区域传送。
- 可规定多于 11 位的同步中断区域接收 (提供 SBF 接收标志位)。

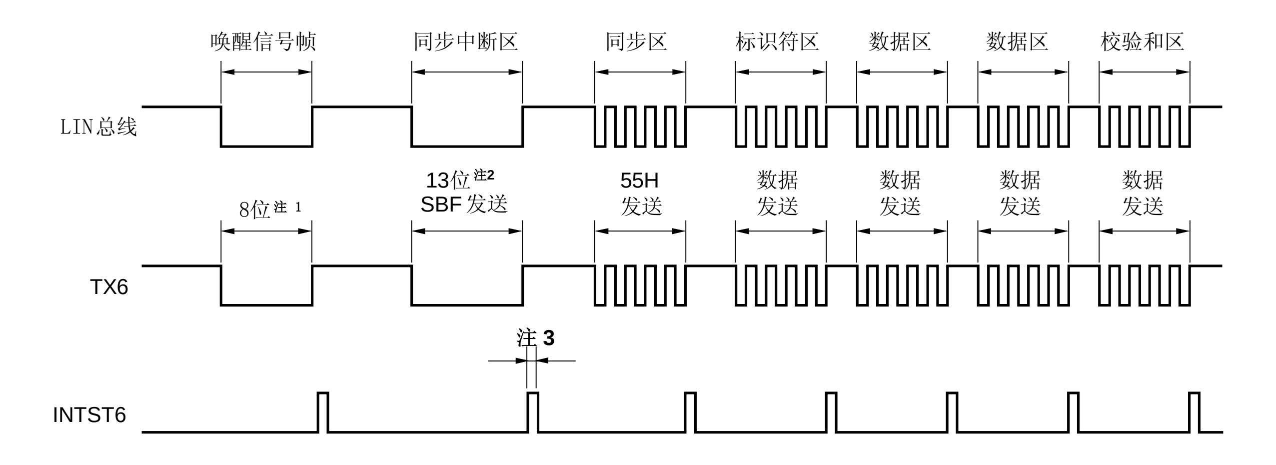
注意事项

1. **TxD6 输出反向功能的操作仅在发送端进行而不在接收端进行。如要使用这个功能，接收端应准备好接收已反向的数据。**
2. **如果串行接口 UART6 使用的时钟未停止（例如，处于 HALT 模式下），则正常操作继续。如果时钟已停止（例如，处于 STOP 模式下），则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。TxD6 引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使 POWER6 = 0，RXE6 = 0，TXE6 = 0。**
3. **如果连续传送数据，从停止位到下一个起始位的通讯时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果在 LIN 模式中，则不能使用连续发送功能。**

备注 LIN 表示局部互联网，是一个低速（1 ～ 20 kbps）串行通信协议，可用于降低网络消耗。
LIN 是一种单主设备（single-master）通信方式，一个主设备上最多可以连接 15 个从设备。
LIN 的从设备用于控制开关，制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。
通常，LIN 主设备与 CAN (控制器局域网 Controller Area Network)等网络相连。
此外，LIN 总线采用单线方式，通过收发器（符合 ISO9141）与各节点相连。
在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率误差。因此当从设备端的波特率误差在±15%范围内，可以进行通信。

图 11-1 和 11-2 概括了 LIN 的发送和接收操作。

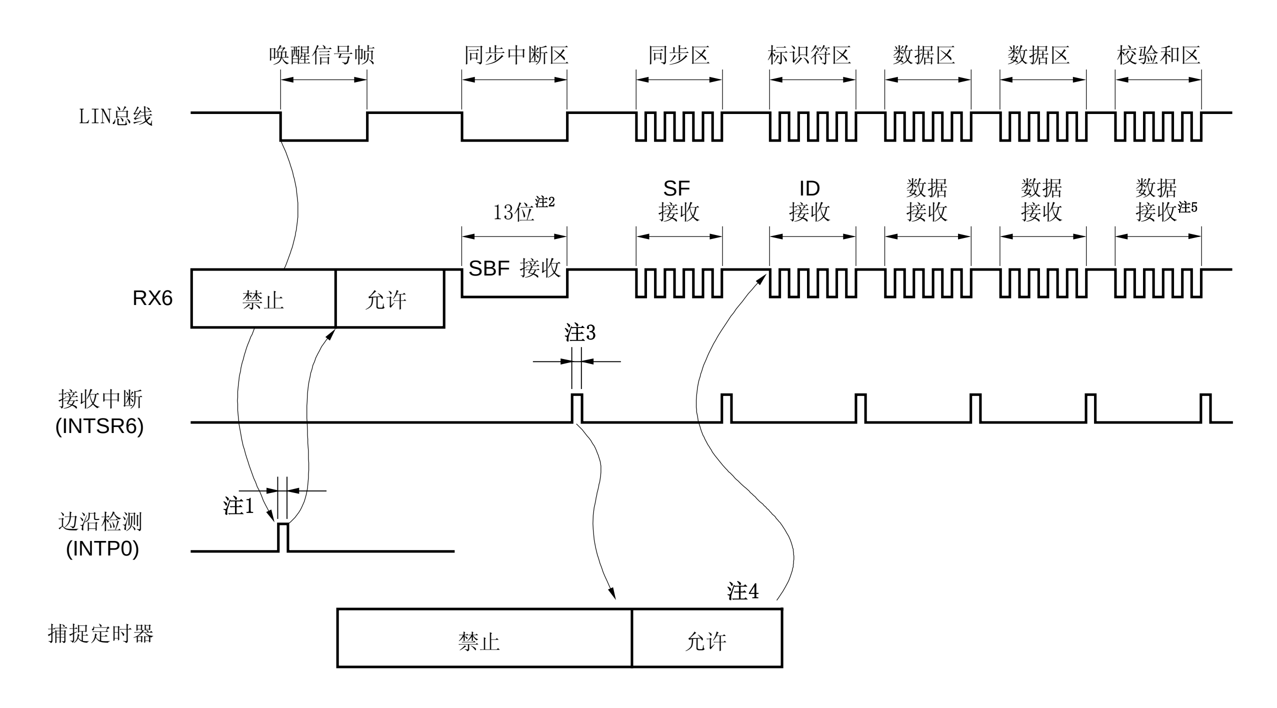
图 11-1. LIN 发送操作



注 1. 在 8 位模式中，可通过发送 80H 替代唤醒信号帧。
2. 使用硬件输出同步中断区。可通过异步串行接口控制寄存器 6 (ASICL6)的 SBL62 ～SBL60 设置输出宽度 (参见 11.4.2 (2) (h) **SBF 发送**)。
3. 发送完成时输出 INTST6。在发送 SBF 后也输出 INTST6。

备注 每个区之前的间隔由软件控制。

图 11-2. LIN 接收操作



注 1. 在引脚的边沿处检测唤醒信号，允许 UART6，并设置 SBF 接收模式。

2. 当检测到停止位时停止接收操作。在检测到 SBF（具有 11 位或更多的低电平数据）时，认为 SBF 接收已正确完成，并输出一个中断信号。在检测到 SBF（具有少于 11 位的低电平数据）时，认为出现 SBF 接收错误，此时不输出中断信号，而是恢复 SBF 接收模式。

3. 如果 SBF 接收已正确完成，将输出一个中断信号。这个 SBF 接收完成中断信号将启动捕捉定时器。禁止 OVE6、PE6 和 FE6 错误检测，不进行 UART 通信的错误检测，也不执行移位寄存器和 RXB6 之间的数据转移。移位寄存器中保存复位值 FFH。

4. 根据同步区的位长度计算波特率误差，在 SBF 接收后禁止使用 UART6，然后重置波特率发生器控制寄存器 6（BRGC6）。

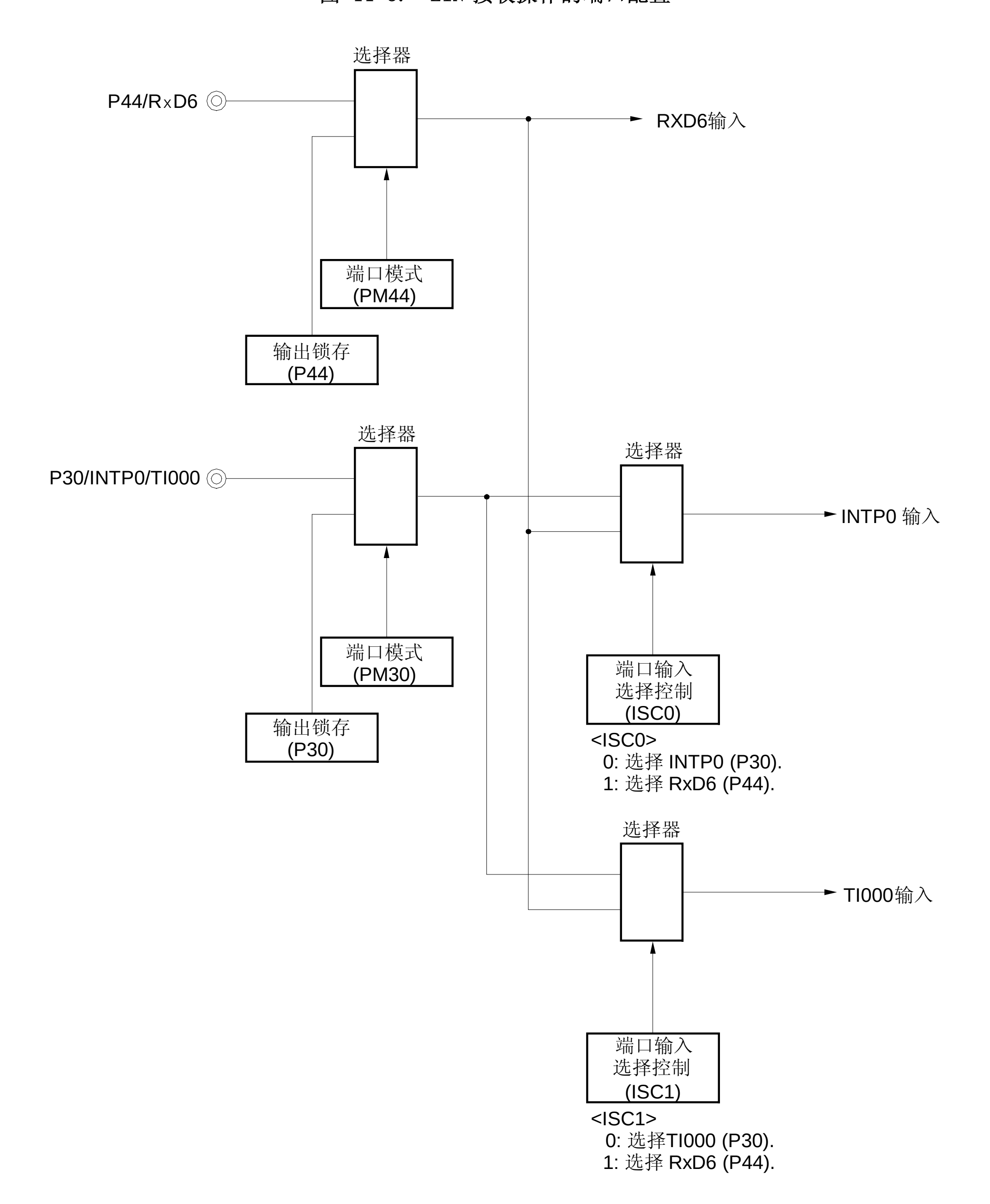
5. 用软件区分校验和区。在接收到校验和区后通过软件对 UART6 进行初始化，并再次通过软件设置 SBF 接收模式。

图 11-3 为 LIN 接收操作的配置。

通过检测外部中断信号(INTP0)脉冲沿，接收从主设备发出的唤醒信号。根据 16 位定时器/事件计数器 00 的外部事件捕捉操作来测量主设备发送的同步区的长度，并计算波特率误差。

不用通过从外部连接 RxD6 和 INTP0/TI000，而由端口输入开关控制(ISC0/ISC1)可将接收端口(RxD6)的输入信号输入给外部中断(INTP0)和 16 位定时器/事件计数器 00。

图 11-3. LIN 接收操作的端口配置



备注　ISC0，ISC1：输入开关控制寄存器(ISC)的第 0 位和第 1 位（参见图 11-11）。

以下是 LIN 通信中用到的外部功能。

<用到的外部功能>

- 外部中断（INTP0）：唤醒信号检测
  用途: 检测唤醒信号的脉冲沿和通信的开始。
- 16 位定时器/事件计数器 00（TI000）：波特率误差检测
  用途: 通过检测同步区(SF)的长度并按位分频，来检测波特率误差(在捕捉模式下测量 TI000 输入脉冲的间隔）。
- 串行接口 UART6

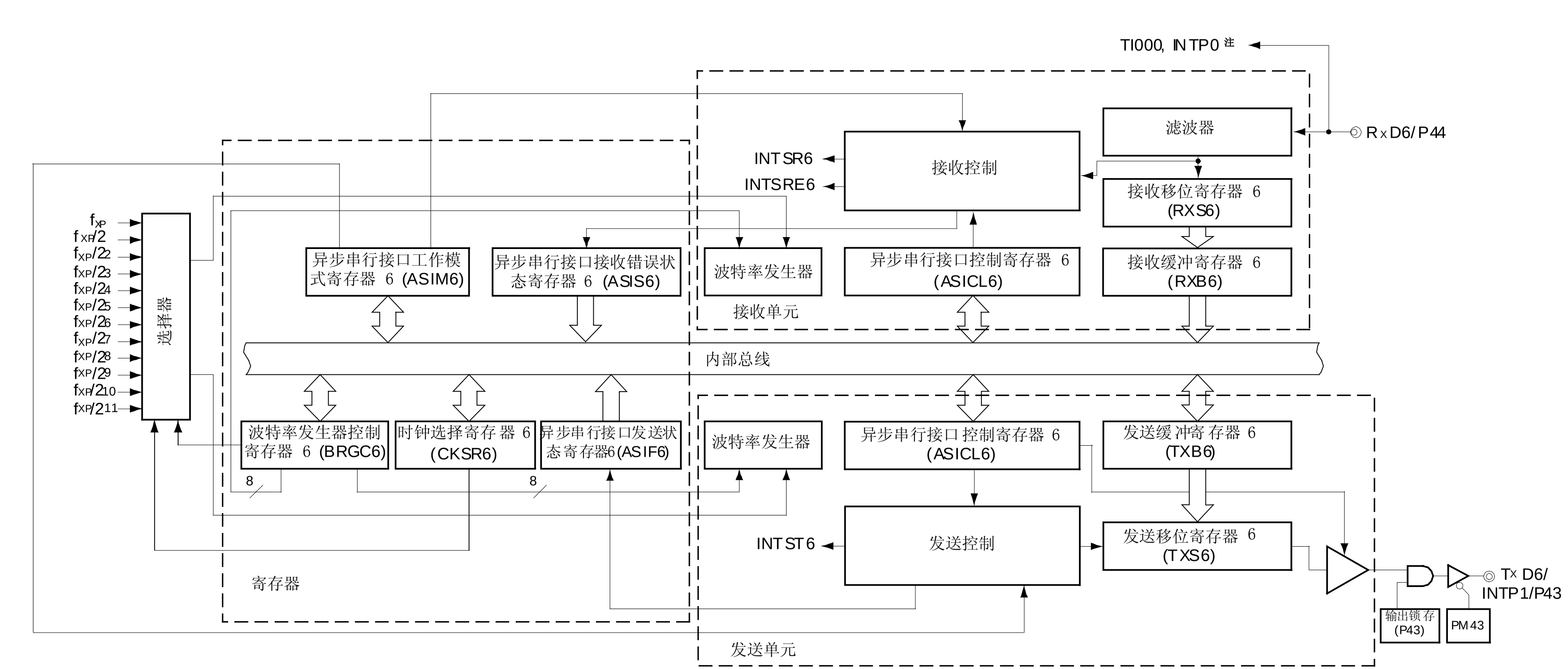
11.2 串行接口 UART6 的配置

串行接口 UART6 由以下硬件组成。

表 11-1. 串行接口 UART6 的配置

项目	配置
寄存器	接收缓冲寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓冲寄存器 6 (TXB6) 发送移位寄存器 6 (TXS6)
控制寄存器	异步串行接口操作模式寄存器 6 (ASIM6) 异步串行接口接收错误状态寄存器 6 (ASIS6) 异步串行接口发送状态寄存器 6 (ASIF6) 时钟选择寄存器 6 (CKSR6) 波特率发生器控制寄存器 6 (BRGC6) 异步串行接口控制寄存器 6 (ASICL6) 输入开关控制寄存器(ISC) 端口模式寄存器 4 (PM4) 端口寄存器 4 (P4)

图 11-4. 串行接口 UART6 的框图



注 可用输入开关控制寄存器(ISC)选择

(1) 接收缓冲寄存器 6 (RXB6)

该 8 位寄存器用于存储由接收移位寄存器(RXS6)转换的并行数据。

每接收到 1 字节的数据，新的接收数据就从接收移位寄存器 6(RXS6)传送到该寄存器中。如果数据宽度被设定为 7 位，则按如下方式发送数据。

- 在 LSB-first 接收模式中，将接收到的数据传送到 RXB6 的 0～6 位，而 MSB 位恒为 0。
- 在 MSB-first 接收模式中，将接收到的数据传送到 RXB6 的 7～1 位，而 LSB 位恒为 0。

如果出现溢出错误(OVE6)，则不把接收数据传送到 RXB6 中。

可由 8 位存储器操作指令读取 RXB6 的内容。不能将数据写入该寄存器中。

复位后该寄存器置 FFH。

(2) 接收移位寄存器 6 (RXS6)

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS6。

(3) 发送缓冲寄存器 6 (TXB6)

该寄存器用于设置发送数据。当数据写入 TXB6 时开始发送。

如果数据宽度被设定为 7 位：

- 在 LSB-first 发送模式中，数据被传送到 TXB6 的 0～6 位，MSB 位不发送。
- 在 MSB-first 发送模式中，数据被传送到 TXB6 的 7～1 位，LSB 位不发送。

可由 8 位存储器操作指令对 TXB6 进行读写操作。

复位后该寄存器置 FFH。

注意事项 1. 当异步串行接口发送状态寄存器 6 (ASIF6)的第 1 位(TXBF6)为 1 时不要对 TXB6 进行写操作。

2. 在通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），不要通过软件刷新（写入相同值）TXB6。

(4) 发送移位寄存器 6 (TXS6)

该寄存器将 TXB6 中要传送的数据作为串行数据从 TXD6 发出。在第 1 次发送过程中当数据写入 TXB6 后立即被传送，或者在持续发送过程中当发出 1 帧数据后，INTST6 中断产生之前数据立即被传送。在基准时钟的下降沿处传送来自 TXB6 的数据，并将其从 TXD6 引脚发出。

程序不能直接对 TXS6 操作。

11.3 控制串行接口 UART6 的寄存器

串行接口 UART6 由以下 9 个寄存器控制。

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入开关控制寄存器(ISC)
- 端口模式寄存器 4 (PM4)
- 端口寄存器 4 (P4)

(1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器用于控制串行接口 UART6 的串行通信。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位后将该寄存器置 01H。

备注 在通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），可以通过软件刷新（写入相同值）ASIM6。

图 11-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (1/2)

地址： FF90H 复位后： 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	允许/禁止内部操作时钟的操作
0 [注 1]	禁止内部操作时钟的操作 (时钟恒为低电平)及对内部电路进行异步复位[注 2]
1 [注 3]	允许内部操作时钟的操作

TXE6	允许/禁止发送
0	禁止发送(同步复位发送电路)
1	允许发送

注
1. 在发送 0 期间，当 POWER6 = 0 时，TXD6 引脚的输出变为高电平，而 RxD6 引脚的输入恒为高电平。
2. 重置异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 6 (ASICL6)的第 7 位(SBRF6) 和第 6 位(SBRT6) ，以及接收缓冲寄存器 6 (RXB6)。
3. POWER6 置 1 后，可在第 2 个基准时钟处进行 8 位计数器的输出操作。

图 11-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (ASIM6) (2/2)

RXE6	允许/禁止接收
0	禁止接收 (同步复位接收电路)
1	允许接收

PS61	PS60	发送操作	接收操作
0	0	不输出校验位.	无校验接收
0	1	输出零校验.	按零校验接收[注]
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL6	指定发送/接收数据的字符长度
0	数据的字符长度 = 7 位
1	数据的字符长度 = 8 位

SL6	指定发送数据停止位的个数
0	停止位的个数 = 1
1	停止位的个数 = 2

ISRM6	允许/禁止出错时接收完成中断的产生
0	出错时产生“INTSRE6” (此时，不产生 INTSR6)
1	出错时产生“INTSR6” (此时，不产生 INTSRE6)

注 如果选择“按零校验接收”，则无法判断校验。因此异步串行接口接收错误状态寄存器 0(ASIS6)的第 2 位(PE6)不会被置 1，这样也不会产生出错中断。

注意事项

1. 启动操作时，对 POWER6 置 1，然后对 TXE6 置 1。要停止操作，先将 TXE6 清零，然后将 POWER6 清零。
2. 启动操作时，对 POWER6 置 1，然后对 RXE6 置 1。要停止操作，先将 RXE6 清零，然后将 POWER6 清零。
3. 当 RxD6 引脚输入为高电平时，先对 POWER6 置 1，再对 RXE6 置 1。当输入为低电平 POWER6 置 1，RXE6 置 1 时，启动接收操作。
4. 在对 PS61、PS60 和 CL6 位重写之前，先将 TXE6 和 RXE6 位清零。
5. 在启动 LIN 设备时 PS61 与 PS60 恒为 0。
6. 在重写 SL6 位时必须确保 TXE6 = 0。当“停止位的个数=1”时，始终执行接收操作，因此接收不会受到 SL6 的影响。
7. 当重写 ISRM6 时，RXE6 必须为 0。

(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)

该寄存器用于指示串行接口 UART6 接收操作完成时出现的错误状态，包括 3 个出错标志位(PE6, FE6, OVE6)。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 5 位(RXE6)、第 7 位(POWER6)均为 0，则复位后将该寄存器清零（00H）。因此当读取该寄存器时读出值为 00H。

图 11-6. 异步串行接口接收错误状态寄存器 6 (ASIS6) 的格式

地址： FF93H 复位后： 00H R

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	指示校验错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0，或 ASIS6 被读取
1	若发送数据的校验位与接收完成时的校验位不匹配

FE6	指示帧错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0，或 ASIS6 被读取
1	若接收完成时未检测到停止位

OVE6	指示溢出错误的状态标志位
0	若 POWER6 = 0 且 RXE6 = 0，或 ASIS6 被读取
1	若将接收数据存于 RXB 寄存器中且在读取该数据前下一个接收操作已完成

注意事项 1. 根据异步串行接口操作模式寄存器 6(ASIM6)的 PS61 和 PS60 位的内容的不同，对 PE6 位的操作也有所不同。

2. 无论停止位的个数是多少，只能将接收到的第 1 位作为停止位进行校验。

3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 6（RXB6）中而是被忽略。

(3) 异步串行接口发送状态寄存器 6 (ASIF6)

该寄存器指示串行接口 UART6 的发送状态，包括两个状态标志位(TXBF6 和 TXSF6)。

当数据从 TXB6 传送到 TXS6 后通过将下一个发送数据写入 TXB6，使得发送能够连续进行而不被中断，即便是在中断期间也不受影响。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 7 位(POWER6)和第 6 位(TXE6)=0，复位后将该寄存器清零(00H)。

图 11-7. 异步串行接口发送状态寄存器 6（ASIF6）的格式

地址: FF95H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓冲数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果数据被传送到发送移位寄存器 6（TXS6）中
1	当数据写入发送缓冲寄存器 6 (TXB6)中(如果数据在 TXB6 中)

TXSF6	发送移位寄存器数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果发送完成后下一个数据没有从发送缓冲寄存器 6（TXB6）发出
1	如果数据从发送缓冲寄存器 6 (TXB6)发出(如果数据仍在发送过程中)

注意事项

1. 如要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 中。检查 TXBF6 标志位，必须为“0”。然后，将下一个发送数据（第二个字节）写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。
2. 若要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

(4) 时钟选择寄存器 6 (CKSR6)

该寄存器用于选择串行接口 UART6 的基准时钟。

可由 8 位存储器操作指令设置 CKSR6。

复位后将该寄存器清零（00H）。

备注 通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），可由软件刷新（写入相同值）CKSR6。

图 11-8. 时钟选择寄存器 6 (CKSR6)的格式

地址： FF96H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	基准时钟 (f_{XCLK6}) 选择
0	0	0	0	f_{XP} (10 MHz)
0	0	0	1	$f_{XP}/2$ (5 MHz)
0	0	1	0	$f_{XP}/2^2$ (2.5 MHz)
0	0	1	1	$f_{XP}/2^3$ (1.25 MHz)
0	1	0	0	$f_{XP}/2^4$ (625 kHz)
0	1	0	1	$f_{XP}/2^5$ (312.5 kHz)
0	1	1	0	$f_{XP}/2^6$ (156.25 kHz)
0	1	1	1	$f_{XP}/2^7$ (78.13 kHz)
1	0	0	0	$f_{XP}/2^8$ (39.06 kHz)
1	0	0	1	$f_{XP}/2^9$ (19.53 kHz)
1	0	1	0	$f_{XP}/2^{10}$ (9.77 kHz)
1	0	1	1	$f_{XP}/2^{11}$ (4.89 kHz)
其它情况				禁止设置

注意事项 当重写 TPS63 ～ TPS60 时，必须确保 POWER6 = 0。

备注 1. 图中括号里的值为 f_X = 10 MHz 时的操作结果。

2. f_{XP}: 外部硬件时钟的振荡频率。

(5) 波特率发生器控制寄存器 6 (BRGC6)

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC6。

复位后将该寄存器置 FFH。

备注 通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），可由软件刷新（写入相同值）BRGC6。

图 11-9. 波特率发生器控制寄存器 6 (BRGC6)的格式

地址： FF97H 复位后： FFH R/W

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	8 位计数器输出时钟选择
0	0	0	0	0	×	×	×	×	禁止设置
0	0	0	0	1	0	0	0	8	$f_{XCLK6}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK6}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK6}/10$
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

注意事项 1. **在重写 MDL67 ～ MDL60 位时，必须确保 ASIM6 的第 6 位(TXE6)和第 5 位(RXE6) = 0。**

2. **波特率为 8 位计数器输出时钟的 1/2。**

备注 1. f_{XCLK6}: 根据 CKSR6 寄存器的 TPS63～TPS60 位选择的基准时钟频率

2. k: 由 MDL67 ～ MDL60 位设置 (k = 8, 9, 10, ..., 255)

3. ×: 不必考虑

(6) 异步串行接口控制寄存器 6 (ASICL6)

该寄存器用于控制串行接口 UART6 的串行通信操作。

可由 1 位或 8 位存储器操作指令设置 ASICL6。

复位后将该寄存器置 16H。

注意事项 通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），可由软件刷新（写入相同值）ASICL6。但是，在 SBF 接收(SBRF6 = 1)或 SBF 发送(在 SBTT6 设置 (1) 与 INTST6 产生之间) 期间，执行刷新操作设置 SBRT6 = 1 且 SBTT = 1，将会再次触发 SBF 接收和 SBF 发送，因此不要设置 SBRT6 = 1 与 SBTT = 1。

图 11-10. 异步串行接口控制寄存器 6 (ASICL6)的格式 (ASICL6) (1/2)

地址： FF98H 复位后： 16H R/W [注]

符号	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF 接收状态标志
0	如果 POWER6 = 0 且 RXE6 = 0 或者如果 SBF 已正确接收
1	SBF 接收操作正在进行中

SBRT6	SBF 接收触发标志
0	–
1	SBF 接收触发

SBTT6	SBF 发送触发标志
0	–
1	SBF 发送触发

注 第 7 位是只读的。

图 11-10. 异步串行接口控制寄存器 6 (ASICL6)的格式 (2/2)

SBL62	SBL61	SBL60	SBF 发送输出宽度控制
1	0	1	SBF 为 13 位长度输出。
1	1	0	SBF 为 14 位长度输出。
1	1	1	SBF 为 15 位长度输出。
0	0	0	SBF 为 16 位长度输出。
0	0	1	SBF 为 17 位长度输出。
0	1	0	SBF 为 18 位长度输出。
0	1	1	SBF 为 19 位长度输出。
1	0	0	SBF 为 20 位长度输出。

DIR6	起始位格式
0	MSB
1	LSB

TXDLV6	允许/禁止 TxD6 反向输出
0	TxD6 普通输出
1	TxD6 反向输出

注意事项 1. 如果出现 SBF 接收错误，则再次返回到 SBF 接收模式下，并保持 SBRF6 标志位不变。
2. 在设置 SBRT6 为 1 之前，必须确保 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)为 1。并且，在设置 SBRT6 为 1 之后，不要在 SBF 接收结束(产生中断请求信号)之前将 SBRT6 清零。
3. SBRT6 的读取值恒为 0。在 SBF 正确接收后 SBRT6 自动清零。
4. 在设置 SBTT6 为 1 之前，必须确保 ASIM6 的第 7 位(POWER6)和第 6 位(TXE6)为 1。并且，在设置 SBTT6 为 1 之后，不要在 SBF 发送结束(产生中断请求信号)之前将 SBTT6 清零。
5. SBTT6 的读取值恒为 0。在 SBF 发送结束时 SBRT6 自动清零。
6. 在重写 DIR6 和 TXDLV6 之前，将 TXE6 和 RXE6 清零。

(7) 输入开关控制寄存器(ISC)

输入开关控制寄存器(ISC)在 LIN 接收期间用于接收从主设备发送的状态信号。
通过设置 ISC 切换输入信号。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位后将该寄存器清零（00H）。

图 11-11. 输入开关控制寄存器 (ISC)的格式

地址: FF8CH　复位后: 00H　R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	TI000 输入源选择
0	TI000 (P30)
1	RxD6 (P44)

ISC0	INTP0 输入源选择
0	INTP0 (P30)
1	RxD6 (P44)

(8) 端口模式寄存器 4 (PM4)

该寄存器可按位设置端口 4 的输入/输出模式。
当 P43/TxD6/INTP1 引脚用于串行接口数据输出时，将 PM43 清零且 P43 的输出锁存置 1。
当 P44/RxD6 引脚用于串行接口数据输入时，将 PM44 置 1。此时 P44 的输出锁存为 0 或 1。
可由 1 位或 8 位存储器操作指令设置 PM4。
复位后将该寄存器置 FFH。

图 11-12. 端口模式寄存器 4 (PM4)的格式

地址: FF24H　复位后: FFH　R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	P4n 引脚 I/O 模式选择 (n = 0 ～ 5)
0	输出模式（输出缓冲打开）
1	输入模式（输出缓冲关闭）

11.4 串行接口 UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口 (UART) 模式

11.4.1 操作停止模式

在此模式下，不能进行串行通信，可降低电源消耗。此外，引脚可作为一般的端口引脚使用。通过将 ASIM6 的第 7、6 和 5 位(POWER6、TXE6 和 RXE6)清零，可设置操作停止模式。

(1) 使用的寄存器

由异步串行接口操作模式寄存器 6(ASIM6)设置操作停止模式。
可由 1 位或 8 位存储器操作指令设置 ASIM6。
复位后将该寄存器置 01H。

地址: FF90H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	允许/禁止内部操作时钟的操作
0[注 1]	禁止内部操作时钟的操作(时钟恒为低电平) 且对内部电路进行异步复位[注 2]。

TXE6	允许/禁止发送
0	禁止发送(同步复位发送电路)

RXE6	允许/禁止接收
0	禁止接收(同步复位接收电路)

注 1. 在发送期间当 POWER6 = 0 时，TXD6 引脚 的输出变成高电平而 RxD6 引脚的输入恒为高电平。
2. 复位异步串行接口接收错误状态寄存器 6(ASIS6)、异步串行接口发送状态寄存器 6(ASIF6)、异步串行接口控制寄存器(ASICL6)的第 7 位(SBRF6)和第 6 位(SBRT6)和接收缓冲寄存器 6(RXB6)。

注意事项 TXE6 和 RXE6 清零后再对 POWER6 清零，可设置操作停止模式。
若要启动操作，设置 POWER6 为 1，然后设置 TXE6 为 1 且 RXE6 为 1。

备注 若将 RxD6/P44 和 TxD6/INTP1/P43 作为通用端口引脚使用，可参见 **第四章 端口功能。**

11.4.2 异步串行接口（UART）模式

此模式中，1 字节数据在传送起始位后发送/接收，可实现全双工操作。

可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 6（ASIM6）
- 异步串行接口接收错误状态寄存器 6（ASIS6）
- 异步串行接口发送状态寄存器 6（ASIF6）
- 时钟选择寄存器 6（CKSR6）
- 波特率发生器控制寄存器 6（BRGC6）
- 异步串行接口控制寄存器 6（ASICL6）
- 输入开关控制寄存器(ISC)
- 端口模式寄存器 4（PM4）
- 端口寄存器 4（P4）

在 UART 模式中设置操作的基本步骤如下。

〈1〉 设置 CKSR6 寄存器（参见图 11-8）。

〈2〉 设置 BRGC6 寄存器（参见图 11-9）。

〈3〉 设置 ASIM6 寄存器的 0～4 位（ISRM6, SL6, CL6, PS60, PS61）（参见图 11-5）。

〈4〉 设置 ASICL6 寄存器的第 0 和 1 位（TXDLV6, DIR6）（参见图 11-10）。

〈5〉 设置 ASIM6 寄存器的第 7 位（POWER6）为 1。

〈6〉 设置 ASIM6 寄存器的第 6 位（TXE6）为 1。 → 允许发送。
设置 ASIM6 寄存器的第 5 位（RXE6）为 1。 → 允许接收。

〈7〉 将数据写入发送缓冲寄存器 6（TXB6）。 → 开始发送数据。

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚的关系如下所示。

表 11-2. 寄存器设置与引脚的关系

POWER6	TXE6	RXE6	PM43	P43	PM44	P44	UART6 操作	引脚功能	
								TxD6/INTP1/P43	RxD6/P44
0	0	0	×[注]	×[注]	×[注]	×[注]	停止	P43	P44
1	0	1	×[注]	×[注]	1	×	接收	P43	RxD6
	1	0	0	1	×[注]	×[注]	发送	TxD6	P44
	1	1	0	1	1	×	发送/接收	TxD6	RxD6

注 可以被设置为端口功能。

备注 ×: 不必考虑

POWER6: 异步串行接口操作模式寄存器 6 (ASIM6)的第 7 位

TXE6: ASIM6 的第 6 位

RXE6: ASIM6 的第 5 位

PM4×: 端口模式寄存器

P4×: 端口输出锁存

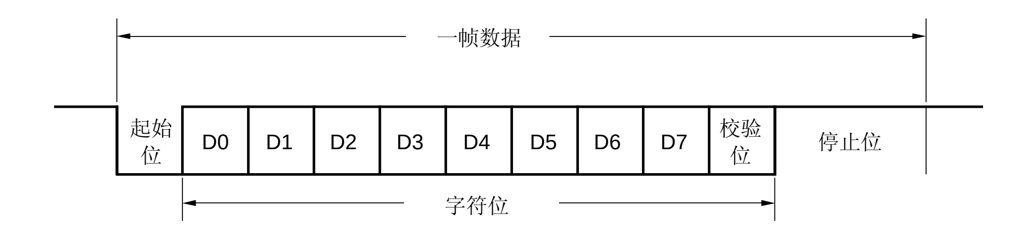
(2) 通信操作

(a) 正常发送/接收数据的格式和波形示例

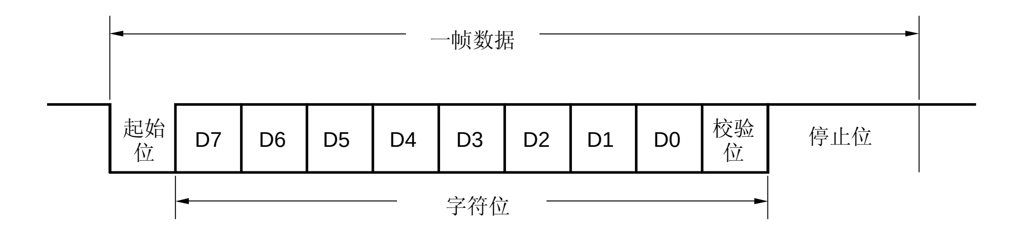
图 11-13 和 11-14 正常发送/接收数据的格式和波形示例。

图 11-13. 正常 UART 发送/接收数据的格式

1. LSB-first 发送/接收



2. MSB-first 发送/接收



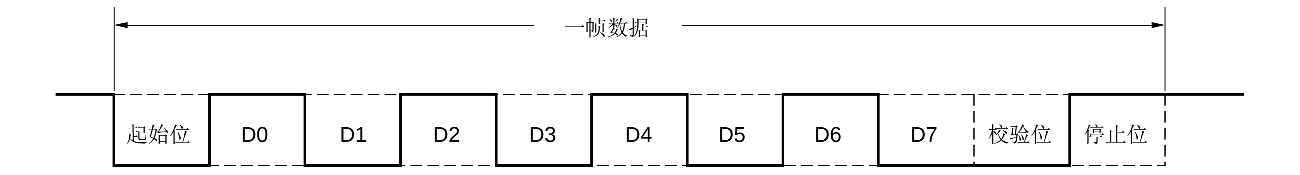
一帧数据由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

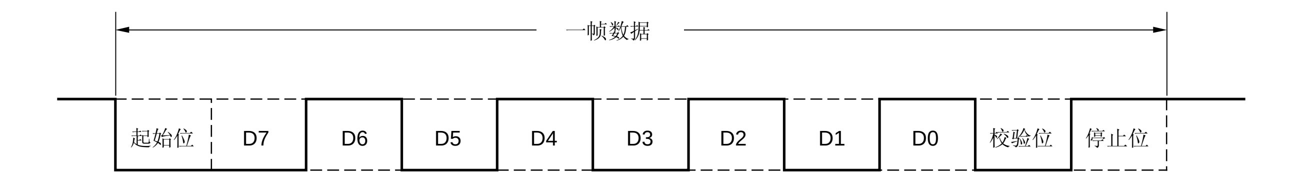
由异步串行接口操作模式寄存器 6(ASIM6)规定一个帧数据中字符位的宽度、校验方式和停止位的宽度。
由异步串行接口控制寄存器 6 (ASICL6)的第 1 位(DIR6)规定数据通信方式(LSB-first 或 MSB-first)。
由 ASICL6 的第 0 位(TXDLV6)规定 TXD6 引脚输出的是正常数据还是反向数据。

图 11-14. 正常 UART 发送/接收数据波形示例

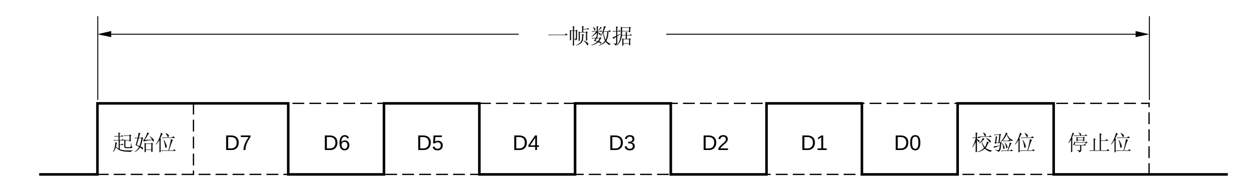
1. 数据长度：8 位，LSB first，校验位：偶校验，停止位：1 位，通信数据：55H



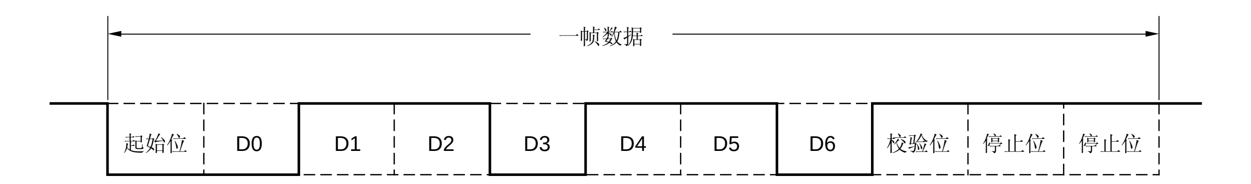
2. 数据长度：8 位，MSB first,校验位：偶校验，停止位：1 位，通信数据：55H



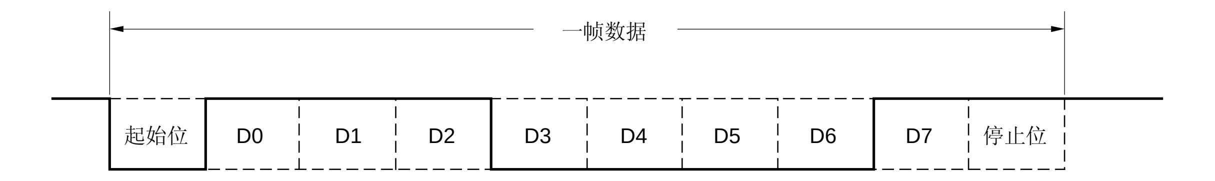
3. 数据长度：8 位，MSB first，校验位：偶校验，停止位：1 位，通信数据：55H，TxD6 引脚输出反向数据



4. 数据长度：7 位，LSB first，校验位：奇校验，停止位：2 位，通信数据：36H



5. 数据长度：8 位，LSB first，校验位：无，停止位：1 位，通信数据：87H



(b) 校验方式与操作

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位（奇数）错误。而采用零校验和无校验时，则不能检测到错误。

注意事项　在 LIN 模式中，PS61 与 PS60 恒为 0。

(i) 偶校验

- 发送

  控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

  校验位的取值如下。

  若发送数据有奇数个“1”：1

  若发送数据有偶数个“1”：0

- 接收

  计算接收数据（包括校验位）中“1”的个数。若“1”的个数为奇数，则产生校验错误。

(ii) 奇校验

- 发送

  控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

  校验位取值如下。

  若发送数据有奇数个“1”：0

  若发送数据有偶数个“1”：1

- 接收

  计算接收数据（包括校验位）中“1”的个数。若“1”的个数为偶数，则产生校验错误。

(iii) 零校验

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

(iv) 无校验

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错误。

(c) 正常发送

当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)为 1 时，TxD6 引脚输出高电平。若 ASIM6 的第 6 位(TXE6) 为 1，则允许发送。将发送数据写入发送缓冲寄存器 6(TXB6)中时，可启动发送操作。起始位、校验位和停止位会自动添加到数据中。

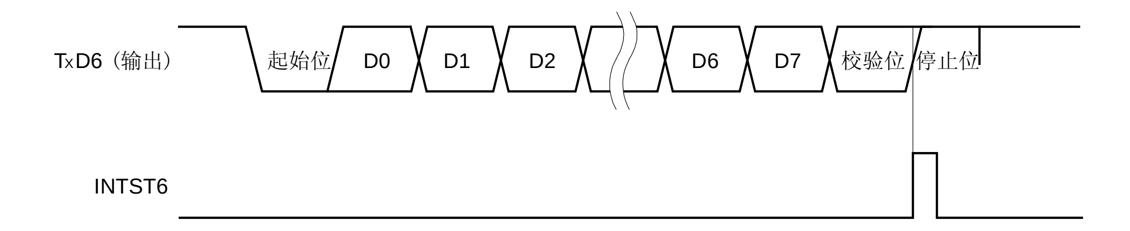
启动发送操作时，先将 TXB6 中的数据传送到发送移位寄存器 6(TXS6)中，然后将数据依次从 TXS6 输出到 TXD6 引脚。当发送结束时，由 ASIM6 设置的校验位和停止位被添加到数据中，并产生发送完成中断请求(INTST6)。

当下一个发送数据写入 TXB6 时，发送操作才结束。

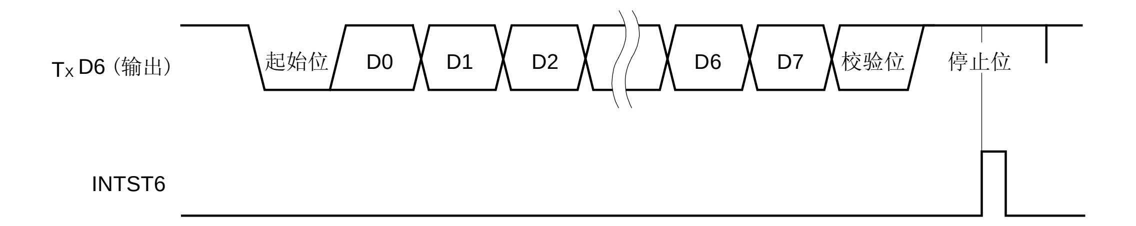
图 11-15 显示了发送完成中断请求(INTST6)的时序。最后一个停止位一输出就产生该中断。

图 11-15. 正常发送完成中断请求时序

1. 停止位长度：1



2. 停止位长度：2



(d) 连续发送

当发送移位寄存器 6 (TXS6)一启动移位操作，下一个发送数据就可以写入发送缓冲寄存器 6 (TXB6)。因此，当发送完一帧数据后，即使在执行 INTST6 中断服务时也能够连续发送数据，从而实现有效通信。此外，当产生发送完成中断时通过读取 ASIF6 的第 0 位（TXSF6），可以对 TXB6 进行两次有效的写操作（2 个字节）而无须等待发送一帧数据的时间。

为了能够连续发送数据，必须根据 ASIF6 检查发送状态和是否可以对 TXB6 进行写操作，然后写入数据。

注意事项 1. 在连续发送期间，ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10” 变成“11”，再变成“01”。因此检查状态时，不能通过比较 TXBF6 和 TXSF6 标志来判断。只能根据读取 TXBF6 标志的值来判断连续发送是否可能。

2. 如果是在 LIN 模式下，则不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6(TXB6)之前，异步串行接口发送状态寄存器(ASIF6) 的值必须为 00H。

TXBF6	写入 TXB6 寄存器
0	允许写入
1	禁止写入

注意事项 若要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 中。检查 TXBF6 标志位，必须为“0”。然后，将下一个发送数据（第二个字节）写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。

可用 TXSF6 标志位检测通信状态。

TXSF6	发送状态
0	发送结束
1	发送正在进行中

注意事项 1. 若要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

2. 在连续发送期间可能会产生错误，即在发送完一帧数据后 INTST6 中断服务执行之前下一个发送操作已完成。可通过编写一段计算发送数据的个数和检查 TXSF6 标志的程序来检测错误。

图 11-16 为连续发送操作流程示例。

图 11-16. 连续发送操作流程示例

设置寄存器

写TXB6

到达传输所需次数？

是

否

读取ASIF6
TXBF6 = 0?

否

是

写TXB6

发送完成中断产生？

否

是

到达传输所需次数？

是

否

读取ASIF6
TXSF6 = 0?

否

是

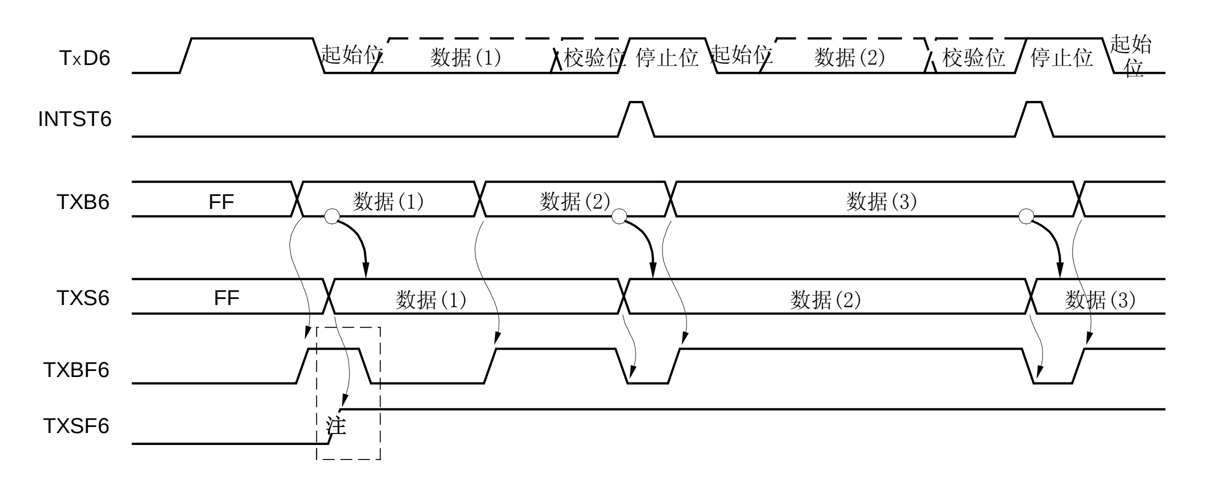
发送完成

备注 TXB6: 发送缓冲寄存器 6
ASIF6: 异步串行接口发送状态寄存器 6
TXBF6: ASIF6 的第 1 位（发送缓冲数据标志）
TXSF6: ASIF6 的第 0 位（发送移位寄存器数据标志）

图 11-17 显示了启动连续发送操作的时序， 图 11-18 显示了结束连续发送操作的时序。



图 11-17. 启动连续发送操作的时序

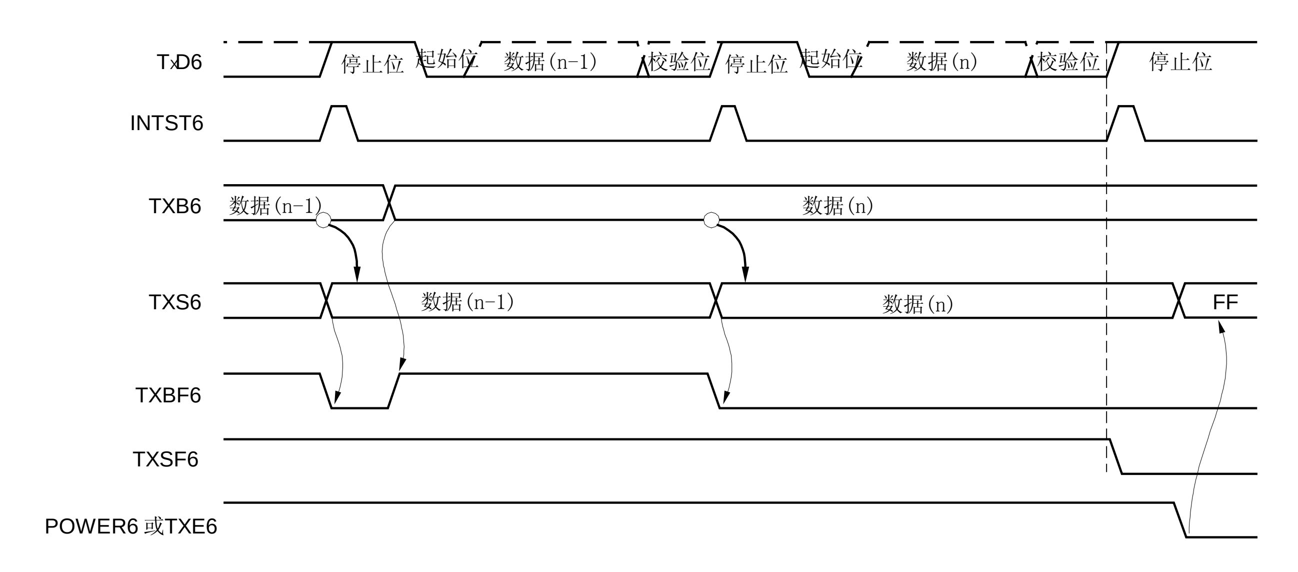


注 在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1, 1 的情况。因此，仅用 TXBF6 判断是否能进行写操作。

备注

TxD6: TxD6 引脚 (输出)
INTST6: 中断请求信号
TXB6: 发送缓冲寄存器 6
TXS6: 发送移位寄存器 6
ASIF6: 异步串行接口发送状态寄存器 6
TXBF6: ASIF6 的第 1 位
TXSF6: ASIF6 的第 0 位

图 11-18. 结束连续发送操作的时序



备注

TxD6:	TxD6 引脚（输出）
INTST6:	中断请求信号
TXB6:	发送缓冲寄存器 6
TXS6:	发送移位寄存器 6
ASIF6:	异步串行接口发送状态寄存器 6
TXBF6:	ASIF6 的第 1 位
TXSF6:	ASIF6 的第 0 位
POWER6:	异步串行接口操作模式寄存器（ASIM6)的第 7 位
TXE6:	异步串行接口操作模式寄存器（ASIM6)的第 6 位

(e) 正常接收

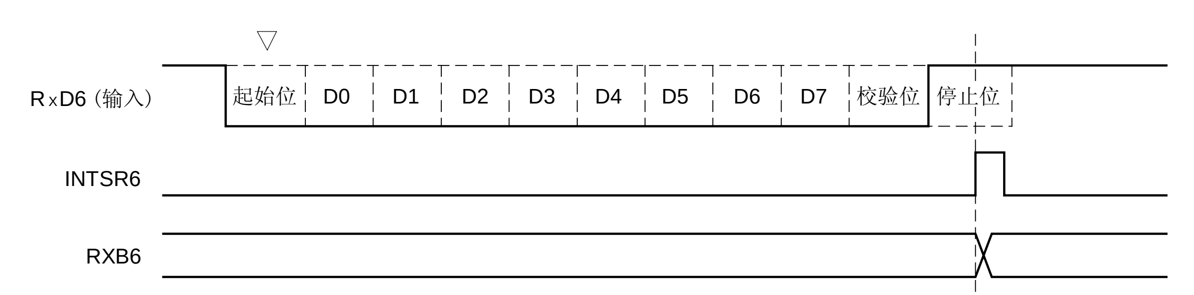
当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)为 1 时，允许接收并对 RxD6 引脚输入进行采样，然后将 ASIM6 的第 5 位(RXE6)置 1。

当检测到 RxD6 引脚输入的下降沿时波特率发生器的 8 位计数器开始计数。当对波特率发生器控制寄存器 6(BRGC6)的值计数时，再次采样 RxD6 的输入信号(如图 11-19 中的▽)。若此时 RxD6 引脚为低电平，则认为是起始位。

当检测到起始位时，开始进行接收操作，以设置的波特率将串行数据依次存入接收移位寄存器 6(RXS6)中。当接收到停止位时，会产生接收完成中断信号(INTSR6)，并且 RXS6 的数据被写入接收缓冲寄存器 6(RXB6)中。但如果出现溢出错误(OVE6)，则不能将接收数据写入 RXB6。

即使在接收过程中出现校验错(PE6)，接收仍然继续，直至接收到停止位，而在接收完成后会产生一个出错中断(INTSR6/INTSRE6)。

图 11-19. 接收完成中断请求时序



注意事项 1. 即使出现接收错误，也必须读取接收缓冲寄存器 6 (RXB6)的内容。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。

2. 接收始终按“停止位的个数为 1”的情况执行，第 2 个停止位被忽略。

3. 在读取 RXB6 之前，必须先读取异步串行接口接收错误状态寄存器 6(ASIS6)。

(f) 接收错误

在接收期间可能会产生三种错误：校验错误，帧错误，或溢出错误。如果异步串行接口接收错误状态寄存器 6 (ASIS6)的错误标志在数据接收后被置位，则会产生接收错误中断请求(INTSR6/INTSRE6)。

可以通过在接收错误中断(INTSR6/INTSRE6)服务程序中读 ASIS6 的内容来识别接收期间发生了哪一个错误(参见 图 11-6)。

当读 ASIS6 时，ASIS6 中的值重置为 0。

表 11-3.　接收错误的原因

接收错误	原因
校验错误	发送与接收数据的校验位不匹配
帧错误	未检测到停止位
溢出错误	在从接收缓冲寄存器 6 (RXB6) 读数据之前下一个数据已接收

根据异步串行接口操作模式寄存器 6 (ASIM6)的第 0 位清零与否，可把错误中断分为接收完成中断(INTSR6)和错误中断(INTSRE6)。

图 11-20.　接收错误中断

1. 如果 ISRM6 被清零(接收完成中断 (INTSR6) 和接收错误中断 (INTSRE6)是分开的)

(a)　接收期间无错误

INTSR6

INTSRE6

(b)　接收期间产生错误

INTSR6

INTSRE6

2. 如果 ISRM6=1 (错误中断包含在 INTSR6 中)

(a)　接收期间无错误

INTSR6

INTSRE6

(b)　接收期间产生错误

INTSR6

INTSRE6

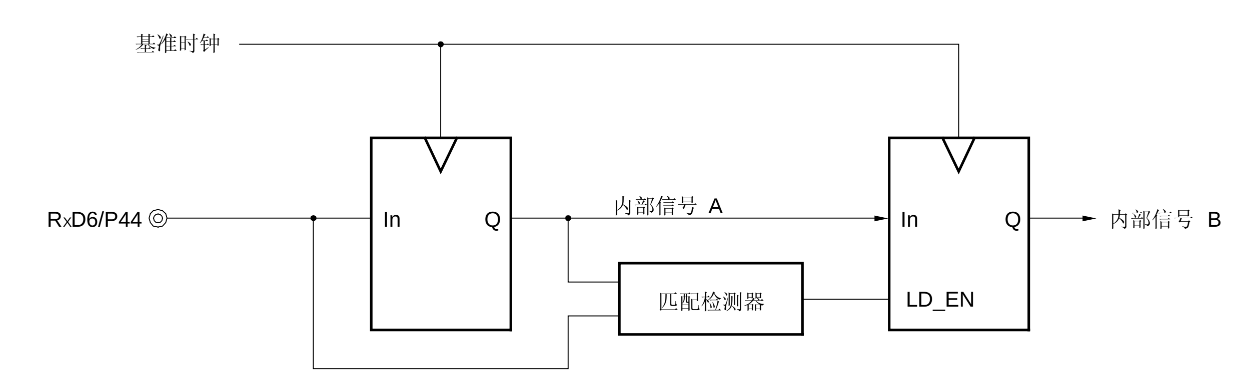
(g) 接收数据的噪声滤波器

使用由预分频模块输出的基准时钟来采样 RxD6 信号。

如果两次采样值相同，则匹配检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声滤波器的电路结构如图 11-21 所示，接收操作的内部过程从外部信号状态开始延迟两个时钟执行。

图 11-21. 噪声滤波器电路



(h) SBF 发送

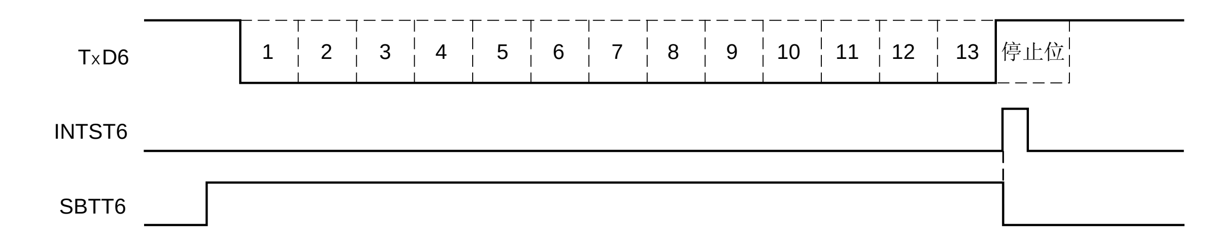
如果在 LIN 模式下，可使用 SBF (同步中断区)发送控制功能进行发送。LIN 的发送操作如**图 11-1 LIN 的发送操作**。

当将 ASIM6 的第 7 位(POWER6)置 1 时，TxD6 引脚输出高电平。然后，当将第 6 位(RXE6)置 1 时，进入发送允许状态，当设置异步串行接口控制寄存器 6(ASICL6)的第 5 位(SBTT6)为 1 时 SBF 发送开始。

此后，第 13～20 位(由 ASICL6 的第 4～2 位(SBL62～SBL60)设置)输出低电平。SBF 发送结束之后，产生发送完成中断请求(INTST6)并且 SBTT6 自动清零。此后，恢复正常发送模式。

发送被挂起直到下一个要发送的数据被写入发送缓冲寄存器 6(TXB6)，或 SBTT6 被置为 1。

图 11-22. SBF 发送



备注 TxD6: TxD6 引脚 (输出)

INTST6: 发送完成中断请求

SBTT6: 异步串行接口控制寄存器 6 (ASICL6)的第 5 位

(i) SBF 接收

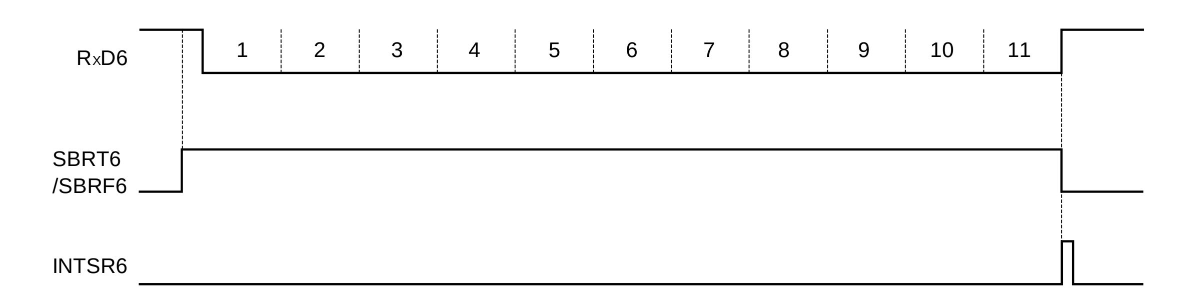
在 LIN 模式下，SBF 接收控制功能用于接收操作。需要了解 LIN 的接收操作，可参见 **图 11-2 LIN 接收操作**。

当将 ASIM6 的第 7 位(POWER6)置 1，然后再将第 5 位(RXE6)置 1 时允许接收。当异步串行接口控制寄存器 6(ASICL6)的第 6 位(SBRT6)为 1 时允许进行 SBF 接收。在 SBF 接收允许状态下，对 RxD6 引脚进行采样，以在正常接收允许状态下同样的方式检测起始位。

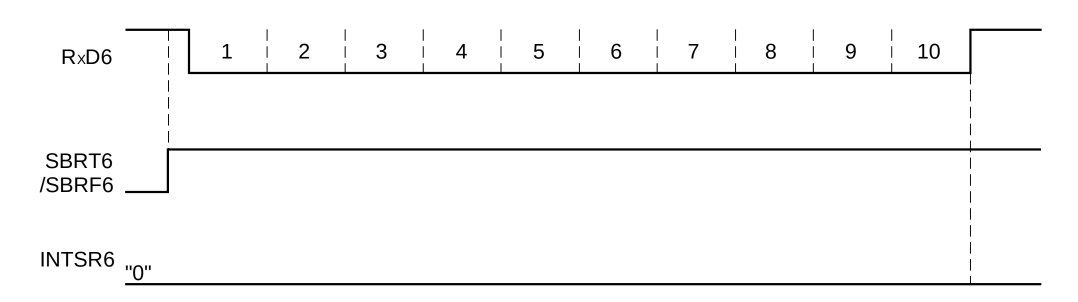
当检测到起始位时，启动接收操作，按设置的波特率将串行数据依次存入接收移位寄存器 6 (RXS6)中。在接收到停止位且 SBF 的宽度为 11 位或更多位时，与正常接收操作相同，将产生接收完成中断请求(INTSR6)。 此时 SBRF6 和 SBRT6 自动清零，SBF 接收操作结束。OVE6、PE6 和 FE6（异步串行接口接收错误状态寄存器 6(ASIS6)的 0～2 位）等错误检测被挂起，不进行 UART 通信的错误检测，也不执行接收移位寄存器 6(RXS6)和接收缓冲寄存器 6(RXB6)的数据传送。移位寄存器中保存复位值 FFH。如果 SBF 的宽度为 10 位或更少位，则在接收到停止位后不会产生出错中断，并恢复 SBF 接收模式。此时不对 SBRF6 和 SBRT6 位清零。

图 11-23. SBF 接收

1. 正常 SBF 接收 (检测到停止位，且 SBF 宽度大于 10.5 位)



2. SBF 接收错误 (检测到停止位，且 SBF 宽度小于或等于 10.5 位)



备注 RxD6: RxD6 引脚 (输入)
SBRT6: 异步串行接口控制寄存器 6 (ASICL6)的第 6 位
SBRF6: ASICL6 的第 7 位
INTSR6: 接收完成中断请求

11.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UART6 发送/接收的串行时钟。
有别于用于发送和接收 8 的 位计数器。

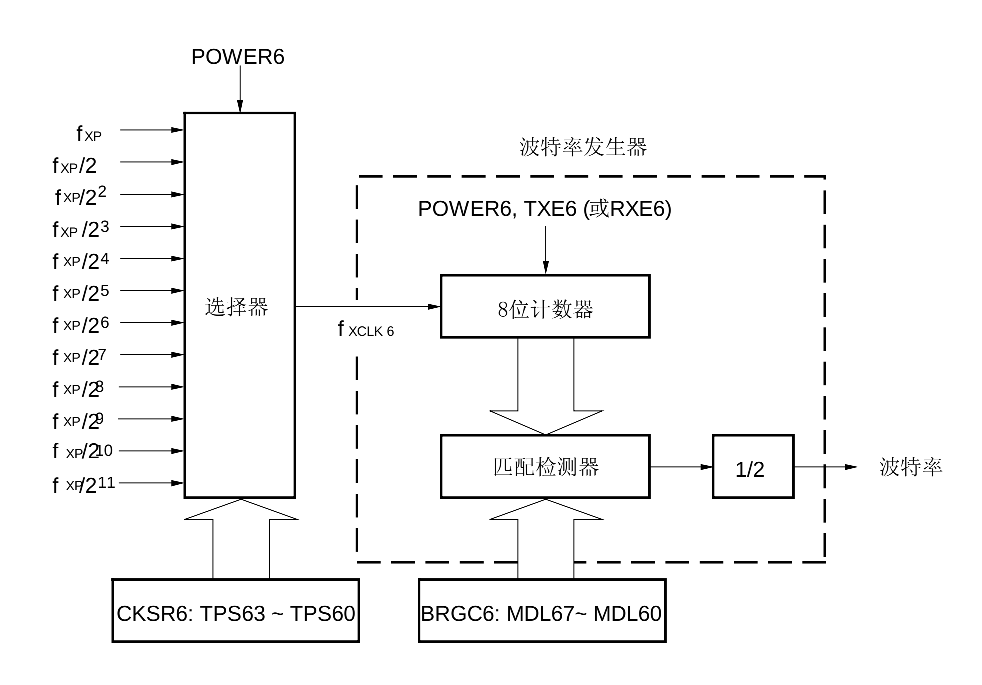
(1) 波特率发生器的配置

- 基准时钟
  当异步串行接口操作模式寄存器 6(ASIM6) 的第 7 位(POWER6)为 1 时，由时钟选择寄存器 6(CKSR6)的 3～0 位(TPS63～TPS60)选择的时钟提供给每个模块。该时钟称为基准时钟，它的频率用 f_{XCLK6} 表示。当 POWER6 = 0 时，基本时钟恒为低电平。

- 发送计数器
  当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)或第 6 位(TXE6)为 0 时，该计数器停止计数，并被清零。
  当 POWER6 = 1 且 TXE6 = 1 时计数器开始计数。
  当第 1 个发送数据写入发送缓冲寄存器 6(TXB6)时，计数器被清零。
  如果连续发送数据，则当发送完 1 帧数据时计数器再次被清零。如果后面没有数据要发送，则计数器不被清零，而是继续计数直至 POWER6 或 TXE6 被清零。

- 接收计数器
  当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)或第 5 位(RXE6)为 0 时，该计数器停止计数，并被清零。
  当检测到起始位时，计数器开始计数。
  在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 11-24. 波特率发生器的配置



备注

POWER6:	异步串行接口操作模式寄存器 6 (ASIM6)的第 7 位
TXE6:	ASIM6 的第 6 位
RXE6:	ASIM6 的第 5 位
CKSR6:	时钟选择寄存器 6
BRGC6:	波特率发生器控制寄存器 6

(2) 串行时钟的产生

使用时钟选择寄存器 6（CKSR6）和波特率发生器控制寄存器 6（BRGC6）可产生串行时钟。

根据 CKSR6 的第 3 ～ 0 位（TPS63 ～ TPS60），可选择输入到 8 位计数器的时钟。

根据 BRGC6 的第 7 ～ 0 位（MDL67 ～ MDL60）选择 8 位计数器的分频值。

(a) 波特率

通过下列公式计算波特率。

- 波特率= $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: 根据 CKSR6 寄存器的 TPS63 ～ TPS60 位选择的基准时钟频率

k: 根据 BRGC6 寄存器的 MDL67 ～ MDL60 位设置(k = 8, 9, 10, ..., 255)

(b) 波特率误差

通过下列公式计算波特率误差。

- 误差(%) = $\left(\frac{\text{实际的波特率 (有误差的波特率)}}{\text{预期的波特率 (正确的波特率)}} - 1 \right) \times 100$ [%]

注意事项 1. **在接收端保持发送期间的波特率误差在一个允许的误差范围内。**

2. **接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。**

举例: 基准时钟频率= 10 MHz = 10,000,000 Hz

设置 BRGC6 寄存器的 MDL67 ～ MDL60 位的值 = 00100001B (k = 33)

目标波特率 = 153600 bps

波特率 = 10 M/(2 × 33)

= 10000000/(2 × 33) = 151,515 [bps]

误差 = (151515/153600 − 1) × 100

= −1.357 [%]

(3) 波特率设置举例

表 11-4. 波特率发生器的数据设置

波特率 [bps]	f_{XP} = 10.0 MHz				f_{XP} = 8.38 MHz				f_{XP} = 4.19 MHz			
	TPS63 ～ TPS60	k	计算值	ERR[%]	TPS63 ～ TPS60	k	计算值	ERR[%]	TPS63 ～ TPS60	k	计算值	ERR[%]
600	6H	130	601	0.16	6H	109	601	0.11	5H	109	601	0.11
1200	5H	130	1202	0.16	5H	109	1201	0.11	4H	109	1201	0.11
2400	4H	130	2404	0.16	4H	109	2403	0.11	3H	109	2403	0.11
4800	3H	130	4808	0.16	3H	109	4805	0.11	2H	109	4805	0.11
9600	2H	130	9615	0.16	2H	109	9610	0.11	1H	109	9610	0.11
10400	2H	120	10417	0.16	2H	101	10371	0.28	1H	101	10475	−0.28
19200	1H	130	19231	0.16	1H	109	19220	0.11	0H	109	19220	0.11
31250	1H	80	31250	0.00	0H	134	31268	0.06	0H	67	31268	0.06
38400	0H	130	38462	0.16	0H	109	38440	0.11	0H	55	38090	−0.80
76800	0H	65	76923	0.16	0H	55	76182	−0.80	0H	27	77693	1.03
115200	0H	43	116279	0.94	0H	36	116389	1.03	0H	18	116389	1.03
153600	0H	33	151515	−1.36	0H	27	155185	1.03	0H	14	149643	−2.58
230400	0H	22	227272	−1.36	0H	18	232778	1.03	0H	9	232778	1.03

备注 TPS63 ～ TPS60: 时钟选择寄存器 6 (CKSR6)的 3 ～ 0 位 (用于设置基准时钟(f_{XCLK6}))

k: 根据 BRGC6 的 MDL67 ～ MDL60 位设置(k = 8, 9, 10, ..., 255)

f_{XP}: 外部硬件的时钟振荡频率

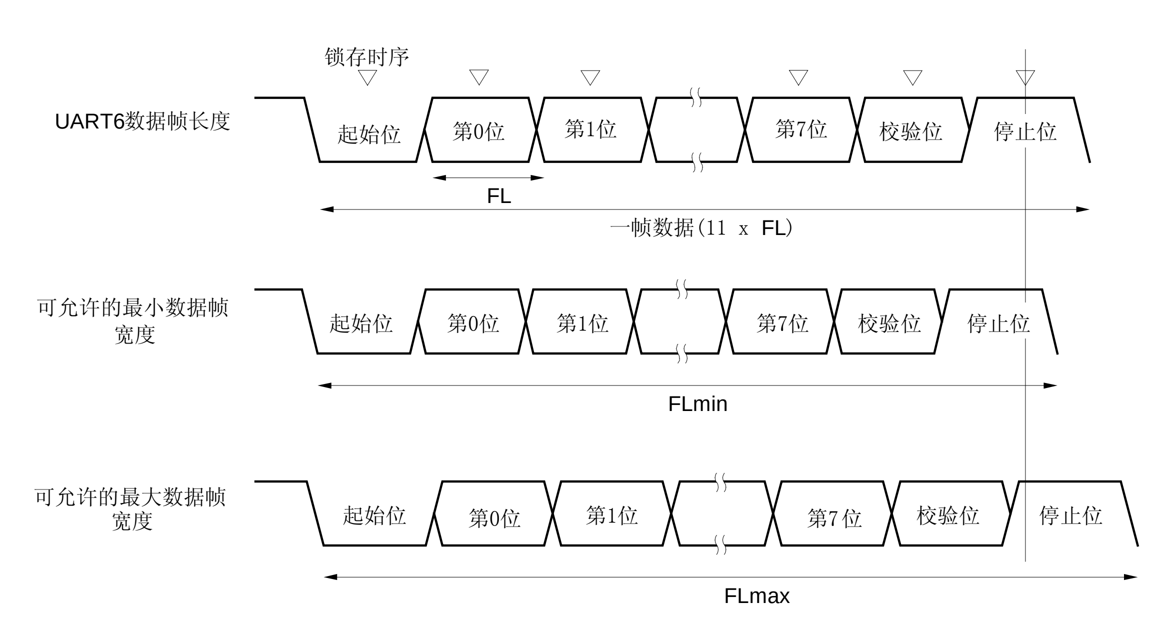
ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 11-25. 接收期间允许的波特率范围



如图 11-25 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 6(BRGC6)设置的计数器确定。如果数据的最后一位 (停止位)满足该锁存时序，则数据可正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$FL = (Brate)^{-1}$

Brate: UART6 的波特率

k: BRGC6 的设置值

FL: 1 位数据宽度

锁存时序的极限: 2 个时钟周期

可允许的最小数据帧宽度：

$$FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此，在发送端设定的可接收的最大波特率如下所示。

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} Brate$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

因此在发送端设定的可接收的最小波特率如下所示。

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} Brate$$

可通过上述最小和最大波特率公式计算在 UART6 与发送端之间允许的波特率误差，如下所示。

表 11-5. 允许的最大/最小波特率误差

分频率(k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	−3.61%
20	+4.26%	−4.31%
50	+4.56%	−4.58%
100	+4.66%	−4.67%
255	+4.72%	−4.73%

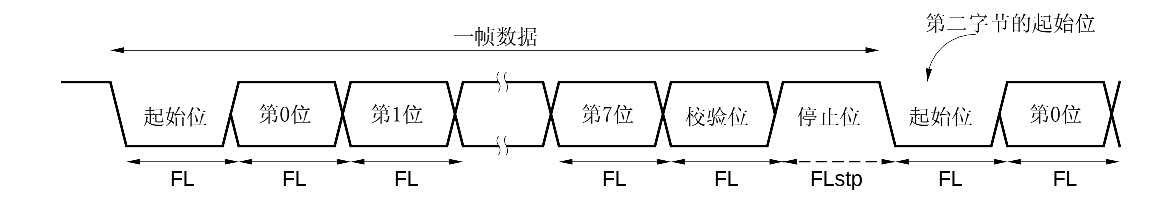
备注 1. 接收时允许的误差与每帧的位数、输入时钟频率和分频率(k)有关。输入时钟频率和分频率(k) 越高，允许的误差就越大。

2. k: BRGC6 的设置值

(5) 连续发送期间数据帧的宽度

当连续发送数据时，从停止位到下一个起始位的数据帧宽度比正常值延长了两个基准时钟周期。不过，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 11-26. 连续发送期间数据帧的宽度



当1 位数据宽度为FL、停止位宽度为FLstp，而基本时钟频率为fXCLK6，有以下公式成立。

$$FLstp = FL + 2/f_{XCLK6}$$

因此，连续发送期间的数据帧宽度为：

$$数据帧宽度 = 11 \times FL + 2/f_{XCLK6}$$

第十二章　乘法器

12.1 乘法器功能

乘法器具有以下功能。

- 计算 8 位 × 8 位 = 16 位

12.2 乘法器配置

(1) 16 位乘法结果存储寄存器 0 (MUL0)

该寄存器存储 16 位乘法结果。

该寄存器在经过 16 个 CPU 时钟后保存乘法结果。

MUL0 能用 16 位存储器操作指令读取。

复位后 MUL0 的值不确定。

注意事项　这个寄存器既能用 16 位存储器操作指令操作，也能用 8 位存储器操作指令操作，使用 8 位存储器操作指令对其进行操作时，须使用直接寻址方式对其进行访问。

(2) 乘法数据寄存器 A 和 B (MRA0 和 MRB0)

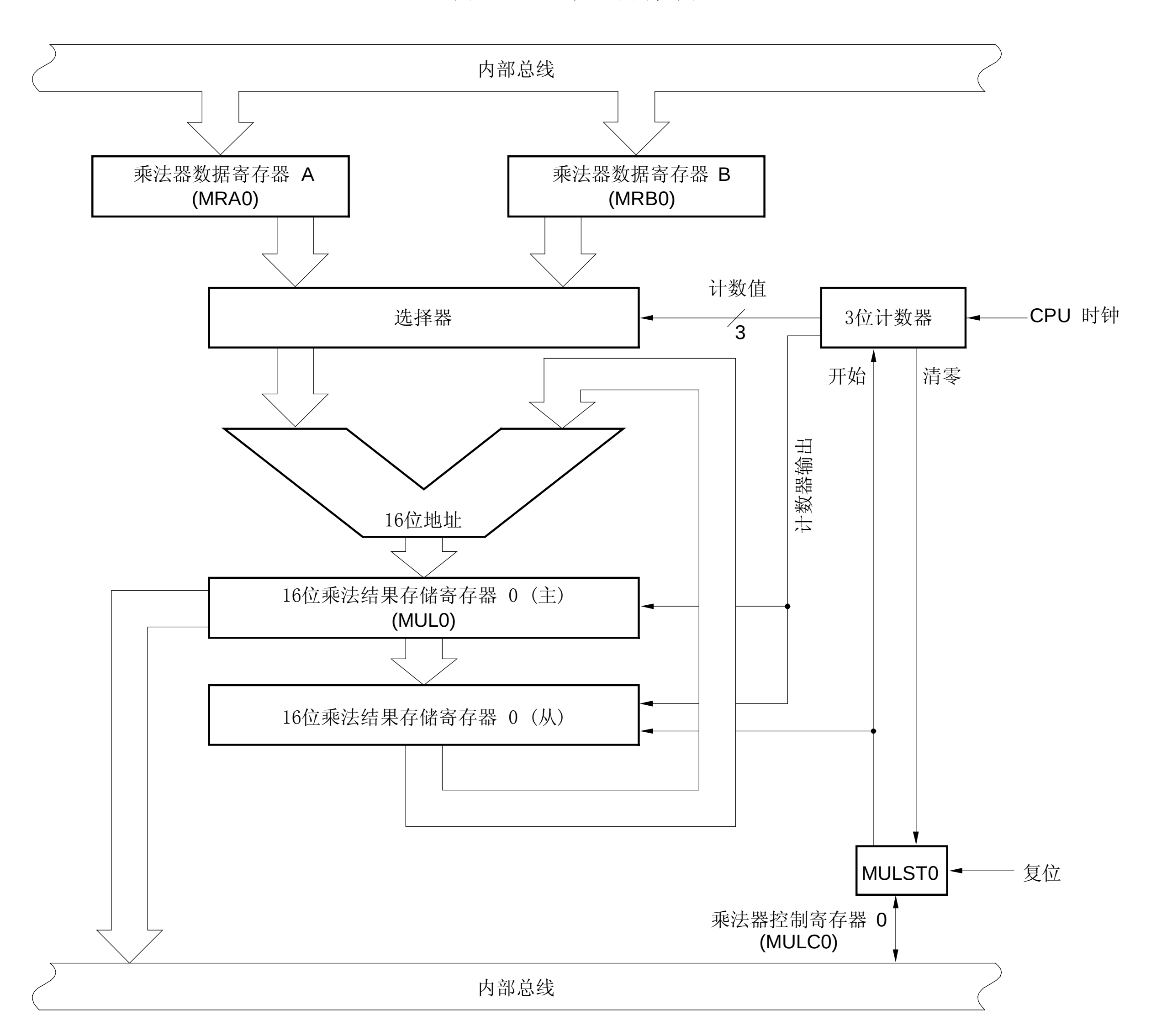
这是 8 位乘法数据存储寄存器。乘法器将 MRA0 和 MRB0 的值进行相乘。

MRA0 和 MRB0 能用 8 位存储器操作指令进行写入。

复位后这两个寄存器的值不确定。

图 12-1 显示了乘法器的框图。

图 12-1.　乘法器的框图


内部总线
乘法器数据寄存器 A
(MRA0)
乘法器数据寄存器 B
(MRB0)
计数值
选择器
3
3位计数器
CPU 时钟
开始
清零
计数器输出
16位地址
16位乘法结果存储寄存器 0（主）
(MUL0)
16位乘法结果存储寄存器 0（从）
MULST0
复位
乘法器控制寄存器 0
(MULC0)
内部总线

12.3 乘法器控制寄存器

乘法器通过下列寄存器进行控制。

- 乘法器控制寄存器 0 (MULC0)

(1) 乘法器控制寄存器 0 (MULC0)

该寄存器指出乘法器操作后的状态，也对乘法器进行控制。

MULC0 能用 1 位或 8 位存储器操作指令进行设置。

复位后该寄存器的值为 00H。

图 12-2.　乘法器控制寄存器 0 (MULC0)的格式

地址：　FFD2H　复位后：　00H　RW

符号	7	6	5	4	3	2	1	<0>
MULC0	0	0	0	0	0	0	0	MULST0

MULST0	乘法器操作开始控制位	乘法器操作状态
0	在计数器重置为 0 后停止操作	操作停止
1	允许操作	操作进行中

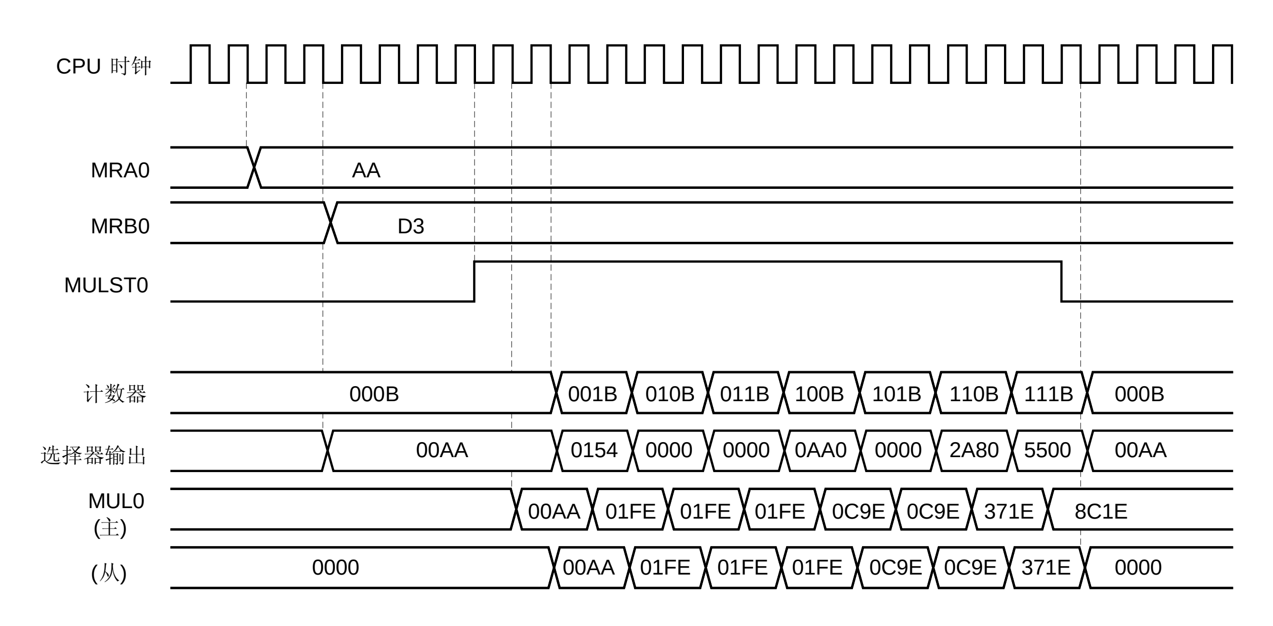
注意事项　一定要将第 1 位到第 7 位清零。

12.4 乘法器操作

78K0S/KB1+ 的乘法器能执行 8 位 × 8 位 = 16 位的计算。图 12-3 显示了当 MRA0 被设置成 AAH，MRB0 被设置成 D3H 时的乘法器操作时序。

〈1〉 通过设置 MULST0 来启动计数。
〈2〉 选择器产生的数据在每个 CPU 时钟与 MUL0 的数据相加，并且计数器的值加 1。
〈3〉 当计数器的值为 111B 时，MULST0 被清零，操作被停止。此时，MUL0 保存结果数据。
〈4〉 当 MULST0 为低时，计数器和从寄存器被清零。

图 12-3.　乘法器操作时序（示例为 AAH × D3H）



第十三章　中断功能

13.1 中断功能类型

所有中断类型都为可屏蔽中断。

- **可屏蔽中断**
  这类中断可进行屏蔽控制。如果两个或更多的中断请求同时发生，将根据中断的优先级，优先响应优先级高的中断。如表 13-1 所示。
  产生一个待机释放信号。
  共有 9 个内部可屏蔽中断源和 4 个外部可屏蔽中断源。

13.2 中断源和配置

共有 13 个可屏蔽中断源，和 4 个复位源（见表 13-1）。

表 13-1. 中断源

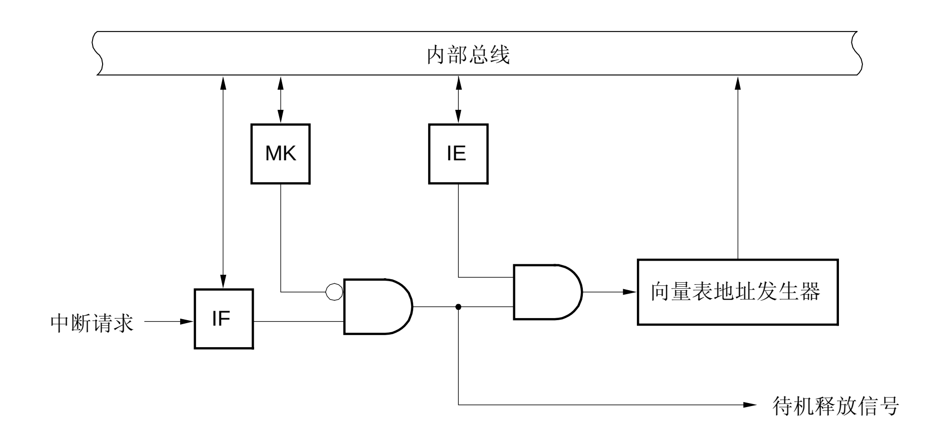
中断类型	优先级[注1]	中断源		内部/外部	向量表地址	基本配置类型[注2]
		名称	触发			
可屏蔽的	1	INTLVI	低电压检测[注3]	内部	0006H	(A)
	2	INTP0	引脚输入边沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTTMH1	TMH1 和 CMP01 相等（指定比较寄存器）	内部	000CH	(A)
	5	INTTM000	TM00 和 CR000 相等(指定比较寄存器)，TI010 引脚有效沿检测（指定捕捉寄存器）		000EH	
	6	INTTM010	TM00 和 CR010 相等(指定比较寄存器)，TI000 引脚有效沿检测（指定捕捉寄存器）		0010H	
	7	INTAD	A/D 转换结束		0012H	
	8	INTP2	引脚输入边沿检测	外部	0016H	(B)
	9	INTP3			0018H	
	10	INTTM80	TM80 和 CR80 相等	内部	001AH	(A)
	11	INTSRE6	UART6 产生接收错误		001CH	
	12	INTSR6	UART6 接收结束		001EH	
	13	INTST6	UART6 发送结束		0020H	
复位	–	RESET	复位输入	–	0000H	–
		POC	上电清零			
		LVI	低电压检测[注4]			
		WDT	WDT 溢出			

注 1. 优先级是指当有多个中断请求输入时各个中断所对应的优先处理顺序。1 是最高优先级，13 为最低优先级。
2. (A)和(B)两种基本配置类型，如图 13-1 中的(A)和(B)所示。
3. 当低电压检测寄存器 (LVIMD) 的第 1 位 (LVIM) = 0 时。
4. 当低电压检测寄存器 (LVIMD) 的第 1 位 (LVIM) = 1 时。

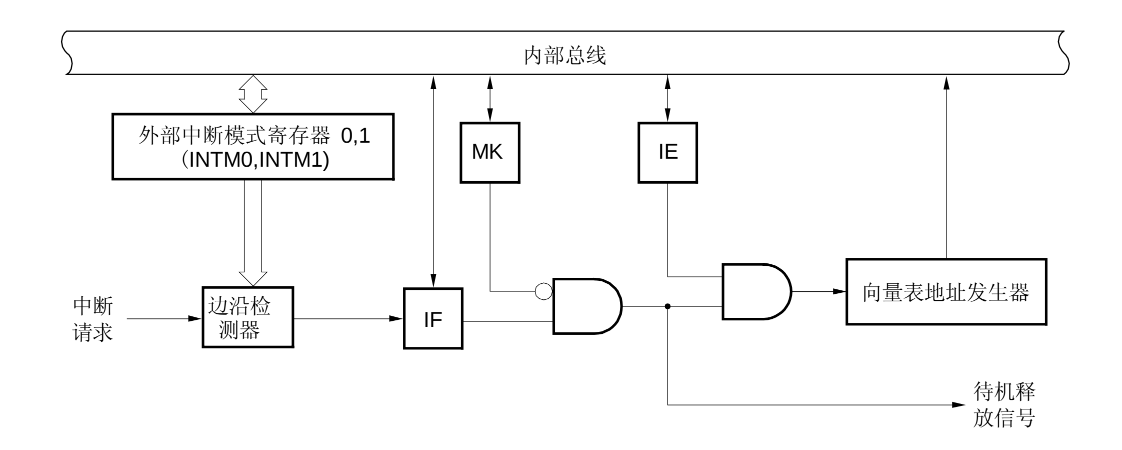
注意事项 向量表地址 0014H 没有相对应的中断源。

图 13-1.　中断功能的基本配置

(A)　内部可屏蔽中断



(B)　外部可屏蔽中断



IF:　中断请求标志
IE:　中断允许标志
MK:　中断屏蔽标志

13.3 控制中断功能的寄存器

以下 4 种寄存器用于控制中断功能。

- 中断请求标志寄存器(IF0, IF1)
- 中断屏蔽标志寄存器(MK0, MK1)
- 外部中断模式寄存器(INTM0, INTM1)
- 程序状态字 (PSW)

表 13-2 列出了中断请求和与其相对应的中断请求标志和中断屏蔽标志。

表 13-2. 中断请求信号及其相应的标志

中断请求信号	中断请求标志	中断屏蔽标志
INTLVI	LVIIF	LVIMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTTMH1	TMIFH1	TMMKH1
INTTM000	TMIF000	TMMK000
INTTM010	TMIF010	TMMK010
INTAD	ADIF	ADMK
INTP2	PIF2	PMK2
INTP3	PIF3	PMK3
INTTM80	TMIF80	TMMK80
INTSRE6	SREIF6	SREMK6
INTSR6	SRIF6	SRMK6
INTST6	STIF6	STMK6

(1)　**中断请求标志寄存器 0，1（IF0，IF1）**

当相应的中断请求产生或执行指令时，中断请求标志被置 1。当中断请求被响应或复位信号产生时，通过执行指令将这些标志清零。

可由 1 位或 8 位存储器操作指令设置 IF0 和 IF1。

复位后将 IF0 和 IF1 清零（00H）。

图 13-2.　中断请求标志寄存器(IF0，IF1)的格式

地址：FFE0H　复位后：00H　R/W

符号	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	0
IF0	ADIF	TMIF010	TMIF000	TMIFH1	PIF1	PIF0	LVIIF	0

地址：FFE1H　复位后：00H　R/W

符号	7	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	0
IF1	0	STIF6	SRIF6	SREIF6	TMIF80	PIF3	PIF2	0

××IF×	中断请求标志
0	没有中断请求信号产生
1	产生中断请求信号；中断请求状态

注意事项　由于 P30，P31，P41，和 P43 有一个复用功能就是作为外部中断输入，所以当指定端口输出模式使输出电平发生变化，中断请求标志置 1。因此，在使用输出模式之前应先将中断屏蔽标志置 1。

(2) 中断屏蔽标志寄存器 0，1 (MK0，MK1)

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断。

可由 1 位或 8 位存储器操作指令设置 MK0 和 MK1 。

复位后将 MK0 和 MK1 设置为 FFH。

图 13-3. 中断屏蔽标志寄存器(MK0，MK1)的格式

地址: FFE4H 复位后: FFH R/W

符号	〈7〉	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	0
MK0	ADMK	TMMK010	TMMK000	TMMKH1	PMK1	PMK0	LVIMK	1

地址: FFE5H 复位后: FFH R/W

符号	7	〈6〉	〈5〉	〈4〉	〈3〉	〈2〉	〈1〉	0
MK1	1	STMK6	SRMK6	SREMK6	TMMK80	PMK3	PMK2	1

××MK×	中断服务控制
0	允许中断服务
1	禁止中断服务

注意事项 由于 P30，P31，P41，和 P43 有一个复用功能就是作为外部中断输入，所以当指定端口输出模式使输出电平发生变化，中断请求标志置 1。因此，在使用输出模式之前应先将中断屏蔽标志置 1。

(3) 外部中断模式寄存器 0（INTM0）

这个寄存器用来设置 INTP0～INTP2 的有效沿。

可用 8 位存储器操作指令设置 INTM0。

复位后使 INTM0 清零（00H）。

图 13-4. 外部中断模式寄存器 0（INTM0）

地址: FFECH　　复位后: 00H　　R/W

符号	7	6	5	4	3	2	1	0
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0

ES21	ES20	INTP2 有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿

ES11	ES10	INTP1 有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿

ES01	ES00	INTP0 有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿

注意事项 1. 第 1 位和第 0 位必须清零。

2. 在设置 INTM0 寄存器前，必须设置相应的中断屏蔽标志（××MK× = 1）以禁止中断。INTM0 寄存器设置完成后，首先对中断请求标志(××IF× = 0)进行清零，然后再对中断屏蔽标志(××MK× = 0)进行清零以允许中断。

(4)　外部中断模式寄存器 1 (INTM1)

INTM1 用来设置 INTP3 的有效沿。

INTM1 可以通过 8 位存储器操作指令进行设置。

复位后使 INTM1 清零（00H）。

图 13-5.　外部中断模式寄存器 1 (INTM1)的格式

地址: FFEDH　　复位后: 00H　　R/W

符号	7	6	5	4	3	2	1	0
INTM1	0	0	0	0	0	0	ES31	ES30

ES31	ES30	INTP3 有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	上升沿和下降沿

注意事项　1. 第 2 位至第 7 位必须清零。

2. 设置 INTM1 之前，先设定 PMK3 为 1 以禁止中断。然后对 PIF3 清 0 再对 PMK3 清 0 允许中断。

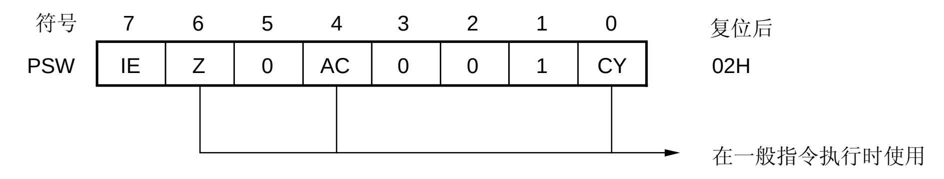
(5)　程序状态字(PSW)

PSW 用于保存指令执行结果和中断请求的当前状态。在 PSW 中包含 IE 标志（允许和禁止可屏蔽中断）。

可用 8 位读写操作指令，还可以用位操作指令和专用指令 (EI 和 DI)对该寄存器进行操作。当一个向量中断被响应时，PSW 自动入栈，IE 标志被清零。

复位后设置 PSW 为 02H。

图 13-6.　程序状态字(PSW) 的格式



IE	是否开启中断
0	关闭
1	开启

13.4 中断服务操作

13.4.1 可屏蔽中断请求的响应操作

当中断请求标志被置 1，相应的中断屏蔽标志被清零时，可以响应可屏蔽中断请求。如果处于中断允许状态（当 IE=1)，向量中断可以被响应。

可屏蔽中断请求产生后开始中断服务所需时间如表 13-3 所示。

图 13-8 和 13-9 为中断请求响应时序。

表 13-3.　从可屏蔽中断请求产生后到开始中断服务所需时间

最小时间	最大时间[注]
9 个时钟周期	19 个时钟周期

注　此处最大时间是指中断请求产生后到 BT 和 BF 指令执行前等待的最大时间。

备注　1 个时钟周期: $\frac{1}{f_{CPU}}$　（f_{CPU}: CPU 时钟）

如果同时产生两个或更多的可屏蔽中断请求，则首先响应优先级高的中断请求。

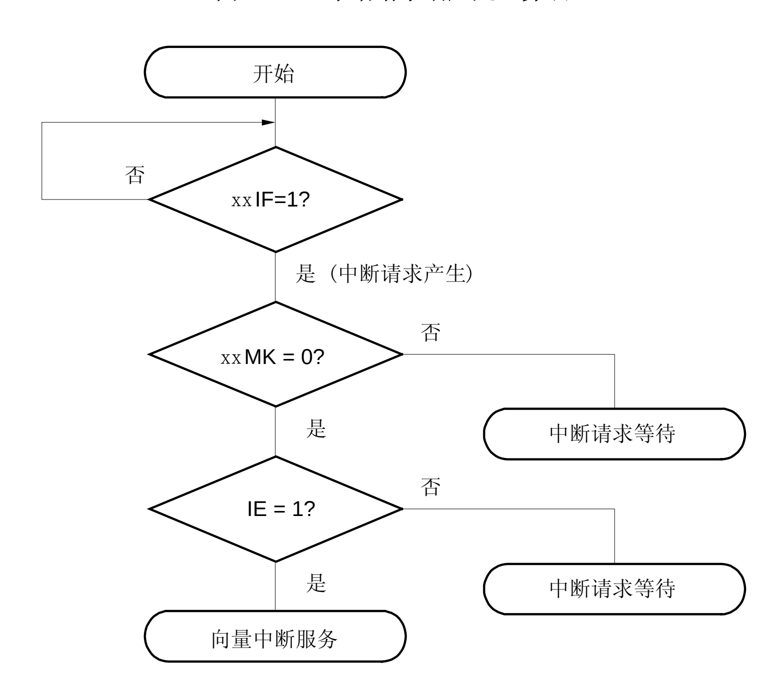
当允许响应中断请求时，处于待机状态中的请求被响应。

图 13-7 显示了中断请求响应算法。

当可屏蔽中断被响应时，PSW 和 PC 的内容被依次保存到堆栈中，IE 标志清零，通过向量表中的数据来判断响应哪个中断请求并执行响应指令。

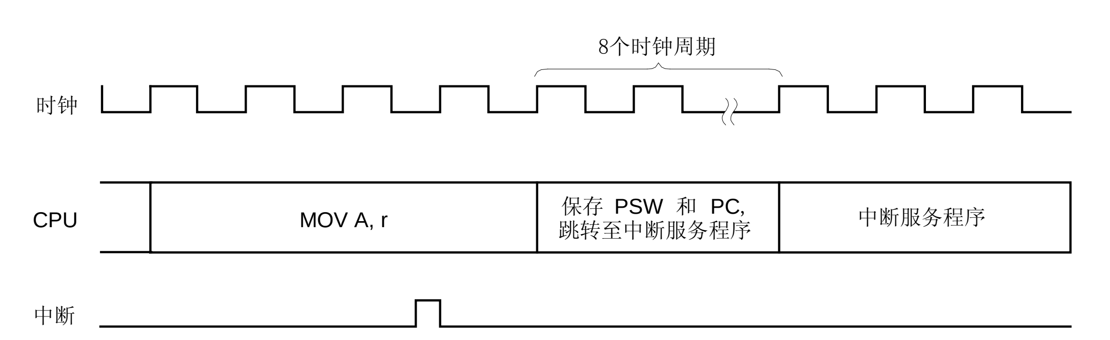
可通过执行 RETI 指令从中断返回。

图 13-7. 中断请求响应处理算法



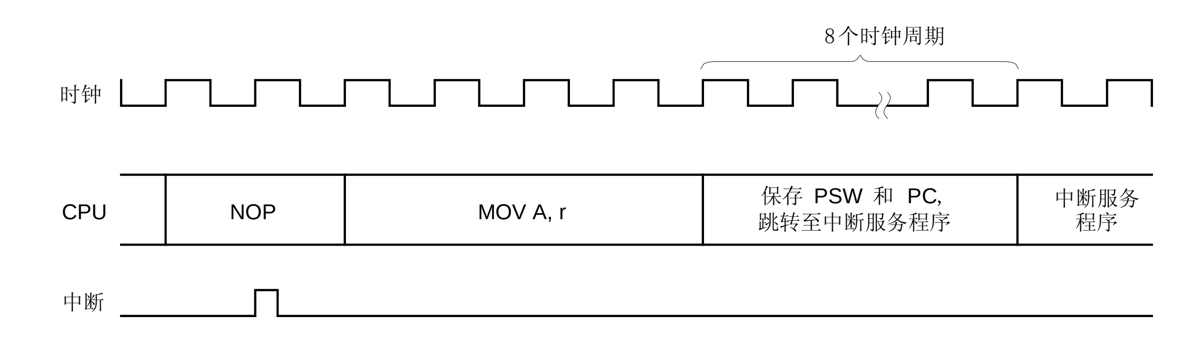
××IF:　中断请求标志
××MK:　中断屏蔽标志
IE:　控制可屏蔽中断请求响应的标志(1 = 允许， 0 =禁止)

图 13-8.　中断请求响应时序（以 MOV A, r 为例）



如果一条指令执行所需的时钟周期为 n（n＝4－10），当执行到 n－1 个时钟周期时中断请求标志（××IF）被置位，那么在该指令执行完成后才能响应中断。图 12－8 为 8 位数据传送指令 MOV A,r 的中断请求响应时序示例。该指令执行时钟周期为 4 个时钟周期，如果中断请求发生在指令执行后的第 3 个时钟周期，中断响应处理在指令 MOV A，r 执行完成后（即第 4 个时钟周期）开始。

图 13-9.　　中断请求响应时序(中断请求标志在指令执行的最后一个时钟周期被置位）



如果中断请求标志在指令执行的最后一个时钟周期被置位，那么中断响应处理在执行下一条指令值之后开始。

图 13-9 为 NOP 指令（2 个时钟周期指令）执行时中断请求响应时序，中断请求标志在第 2 个时钟周期被置位，NOP 指令的下一条指令为 MOV A,r，中断请求响应处理在执行指令 MOV A,r 时开始。

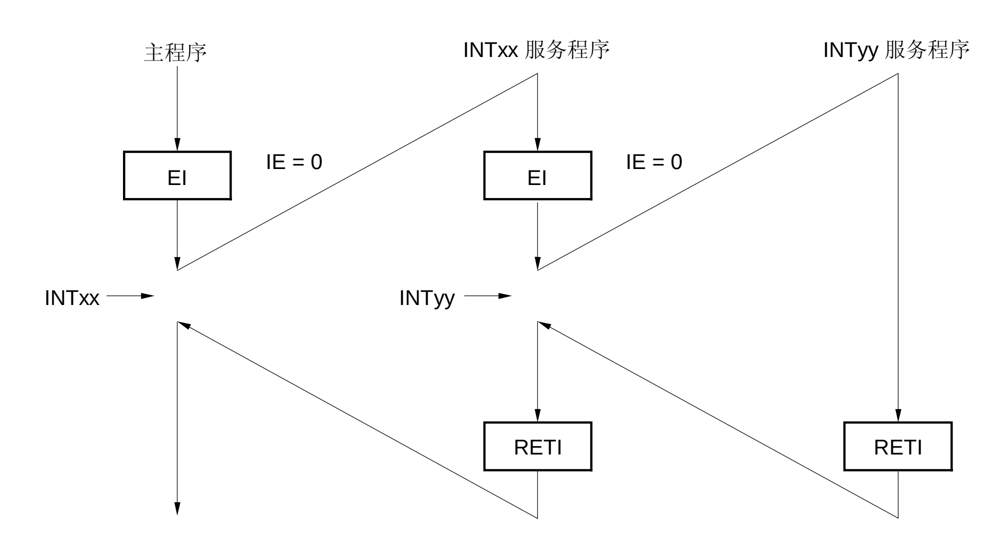
注意事项 当中断请求标志寄存器（IF0,IF1）或中断屏蔽标志寄存器（MK0,MK1）被访问期间，中断请求等待。

13.4.2 中断嵌套

中断嵌套是指在一个中断服务程序进行的同时使用优先级顺序系统响应另一个中断。当两个或者更多中断同时产生时，根据为每个中断提前指定的优先级执行中断服务程序。(见表 13-1).

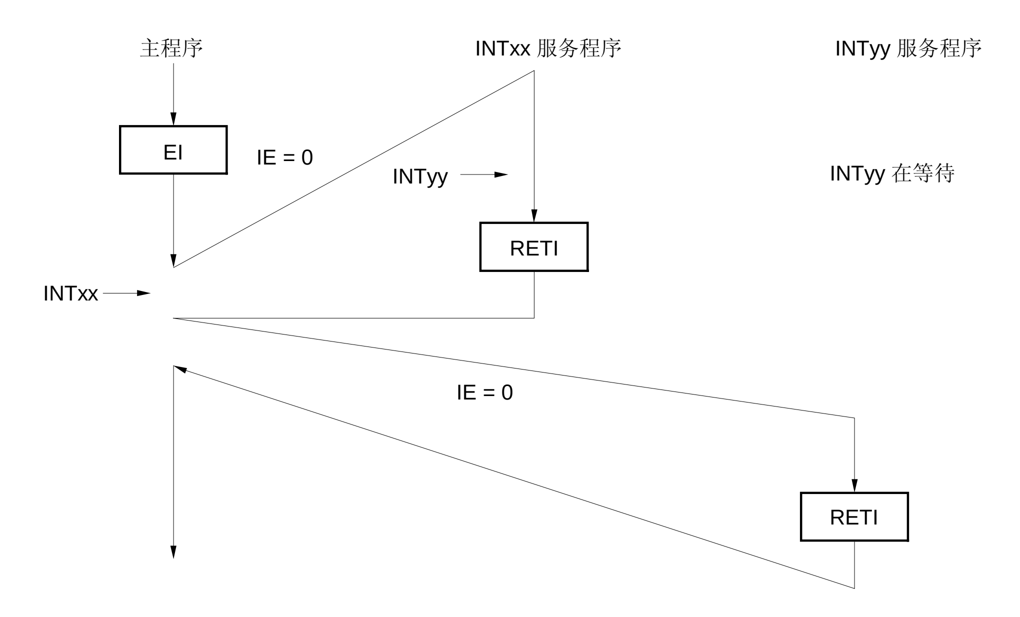
图 13-10. 中断嵌套示例

例 1. 响应中断嵌套



在中断 INTxx 处理期间，中断请求 INTyy 被响应，发生中断嵌套。在每个中断响应之前执行 EI 指令，并且设置中断请求响应为允许状态。

例 2.　由于不允许中断没有产生中断嵌套



在 INTxx 中断服务期间不允许中断（不执行 EI 指令），因此不响应中断请求 INTyy，这样也不会产生中断嵌套。中断请求 INTyy 处于等待状态，在 INTxx 中断服务程序完成后被响应。

IE = 0: 禁止响应中断请求

13.4.3 中断请求等待

在执行某些指令时，如果有中断请求发生（可屏蔽中断或者外部中断），应等执行完此条指令和它的下一条指令后才响应中断，此期间中断响应暂时等待。以下为中断请求等待指令。

- 对中断请求标志寄存器(IF0, IF1)进行操作
- 对中断屏蔽标志寄存器(MK0, MK1)进行操作

第十四章　待机功能

14.1 待机功能及配置

14.1.1 待机功能

表 14-1.　每种工作状态中操作时钟之间的关系

<table>
<tr><td rowspan="3">状态
工作模式</td><td colspan="3">低速 Ring-OSC 振荡器</td><td rowspan="3">系统时钟</td><td rowspan="3">供给外部硬件的时钟</td></tr>
<tr><td rowspan="2">注 1</td><td colspan="2">注 2</td></tr>
<tr><td>LSRSTOP = 0</td><td>LSRSTOP = 1</td></tr>
<tr><td>复位</td><td colspan="3">停止</td><td rowspan="2">停止</td><td rowspan="2">停止</td></tr>
<tr><td>STOP</td><td rowspan="2">振荡</td><td rowspan="2">振荡[注 3]</td><td rowspan="2">停止</td></tr>
<tr><td>HALT</td><td>振荡</td><td>振荡</td></tr>
</table>

注　1. 当由可选字节选择“不能被停止”低速 Ring-OSC 振荡器时。

2. 当选择了低速 Ring-OSC 振荡器“可通过软件停止”时，低速 Ring-OSC 振荡器的振荡可被 LSRSTOP 停止。

3. 如果看门狗定时器的操作时钟为低速 Ring-OSC 时钟，看门狗定时器被停止。

注意事项　仅当由可选字节选择低速 Ring-OSC 振荡器“可通过软件停止”时，LSRSTOP 设置才有效。

备注　LSRSTOP:　低速 Ring-OSC 模式寄存器 (LSRCM)的第 0 位。

待机功能用于减少系统的工作电流，有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式下，CPU 操作时钟停止。系统时钟振荡器继续振荡。如果在设置 HALT 模式之前低速 Ring-OSC 振荡器在工作，则低速 Ring-OSC 振荡器继续振荡(参见表 14-1.　由可选字节选择低速 Ring-OSC 时钟(是否能通过软件停止))。在此模式中，工作电流不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和进行间断的操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，系统时钟振荡器停止操作，整个系统的停止操作，这样 CPU 的工作电流将会大幅下降。

可通过中断请求释放该模式，程序可以间断执行。

但是，由于在 STOP 模式释放后需要一段等待时间[注]以确保振荡器振荡稳定(当使用陶瓷/晶体振荡器时)，因此如果需要响应中断请求后程序立即运行，则应采用 HALT 模式。

注　操作停止时间为 17μs(最小),34μs(典型值),和 67μs(最大)。

在这两种模式中，寄存器、标志和数据存储器的内容将会保持进入待机模式前的内容。I/O 端口输出锁存和输出缓冲状态也将保持。

注意事项
1. **当改变到 STOP 模式时，一定要在执行 STOP 指令(除了以低速 Ring-OSC 时钟工作的外部硬件之外)之前停止外部硬件操作。**
2. **当使用待机功能时推荐按以下步骤减少 A/D 转换器的工作电流：首先对 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)和第 0 位 (ADCE)清零以停止 A/D 转换，然后执行 HALT 或 STOP 指令。**
3. **如果在置 STOP 模式之前低速 Ring-OSC 振荡器正在工作，则在 STOP 模式中不会停止低速 Ring-OSC 时钟 (参见 表 14-1)。**

14.1.2 待机期间用到的寄存器

待机模式释放后的振荡稳定时间由振荡稳定时间选择寄存器(OSTS)控制。

备注　　有关这些寄存器启动、终止以及时钟选择的情况，可参见 **第五章 时钟发生器。**

(1)　振荡稳定时间选择寄存器 (OSTS)

此寄存器用来选择 STOP 模式释放时振荡器时钟的振荡稳定时间。仅当选择晶体/陶瓷振荡时钟作为系统时钟且释放 STOP 模式后，由 OSTS 设置的等待时间才有效。如果选择高速 Ring-OSC 振荡或外部时钟输入作为系统时钟源，不用等待时间。

由可选字节选择系统时钟振荡器和上电或复位释放后经过的振荡稳定时间。详情可参见**第十八章 可选字节**。

可由 8 位存储器操作指令设置 OSTS。

图 14-1.　振荡稳定时间选择寄存器 (OSTS) 的格式

地址： FFF4H　　复位后： 不确定　　R/W

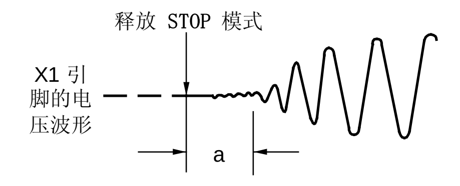
符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	0	OSTS1	OSTS0

OSTS1	OSTS0	振荡稳定时间的选择
0	0	2^{10}/fx (102.4 μs)
0	1	2^{12}/fx (409.6 μs)
1	0	2^{15}/fx (3.27 ms)
1	1	2^{17}/fx (13.1 ms)

注意事项 1.　如要设置并释放 STOP 模式，按以下方式设置振荡稳定时间。

预期振荡稳定时间 ≤ OSTS 设置的振荡稳定时间

2.　无论是复位后还是由中断产生释放 STOP 模式，释放 STOP 模式后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间 (即下图“a” 所示的部分)。



3.　由可选字节选择上电或复位释放后经过的振荡稳定时间。详情可参见第十八章 可选字节。

备注　1.　( )： fx = 10 MHz

2.　通过检查使用的振荡器的特性确定振荡器的振荡稳定时间。

14.2 待机功能的操作

14.2.1 HALT 模式

(1) HALT 模式

执行 HALT 指令设置 HALT 模式。

HALT 模式中的工作状态如下所示。

注意事项 由于中断请求信号可用来释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即释放待机模式。

表 14-2. HALT 模式中的工作状态

项目 \ 模式设置 \ HALT		低速 Ring-OSC 振荡器不能被停止[注]	低速 Ring-OSC 振荡器可被停止[注]	
			当低速 Ring-OSC 振荡继续	当低速 Ring-OSC 振荡停止
系统时钟		CPU 时钟停止		
CPU		操作停止		
端口（锁存）		保持设置 HALT 模式前的状态		
16 定时器/事件计数器 00		可操作		
8 位定时器 80		可操作		
8 位定时器 H1	设置计数时钟为 f_{XP} 到 $f_{XP}/2^{12}$	可操作		
	设置计数时钟为 $f_{RL}/2^{7}$	可操作	可操作	操作停止
看门狗定时器	选择系统时钟作为操作时钟	禁止设置	操作停止	
	选择低速 Ring-OSC 时钟作为操作时钟	可操作 (操作继续)	操作停止	
A/D 转换器		可操作		
串行接口 UART6		可操作		
POC 电路		可操作		
低电压检测		可操作		
外部中断		可操作		

注 由可选字节可选择低速 Ring-OSC 振荡器“不能被停止”或“通过软件停止”（可选字节可参见 **第十八章 可选字节**）。

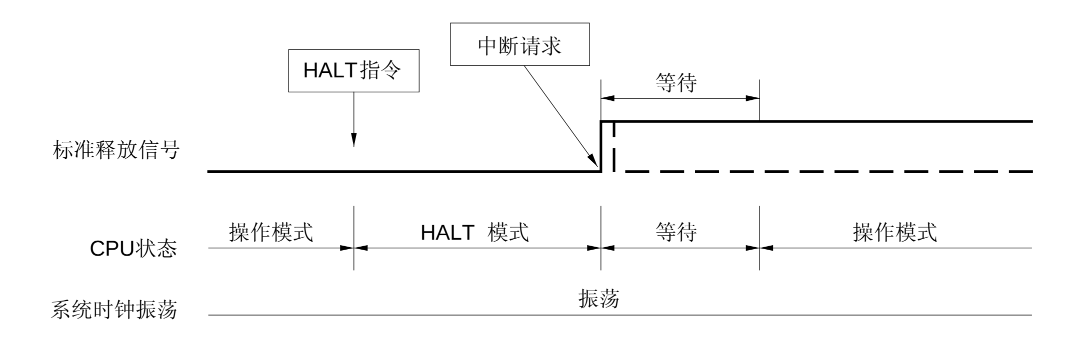
(2) 释放 HALT 模式

可由以下两种中断源释放 HALT 模式。

(a) 由未屏蔽的中断请求释放

当产生一个未屏蔽的中断时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 14-2. 通过产生中断请求释放 HALT 模式



备注　1. 虚线表示释放待机模式的中断请求被响应的情况。

2. 等待时间如下所示:

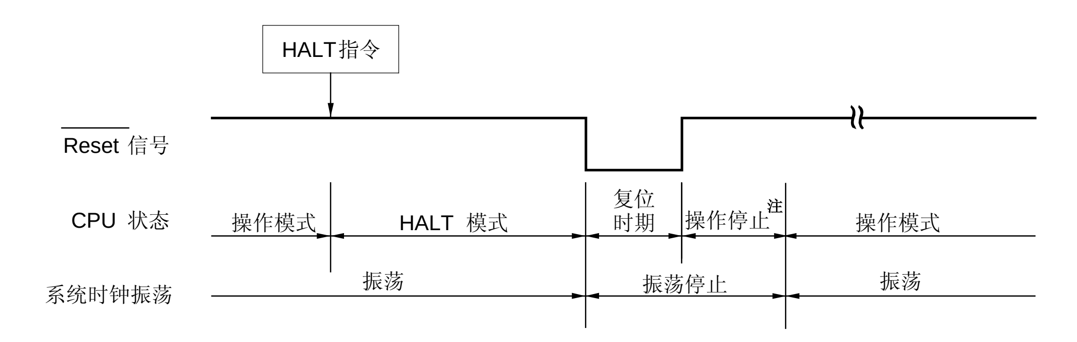
- 执行向量中断服务程序时:11～13 个时钟周期
- 不执行向量中断服务程序时:3～5 个时钟周期

(b) 由复位信号释放

当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作后，程序从复位向量指向的地址处执行。

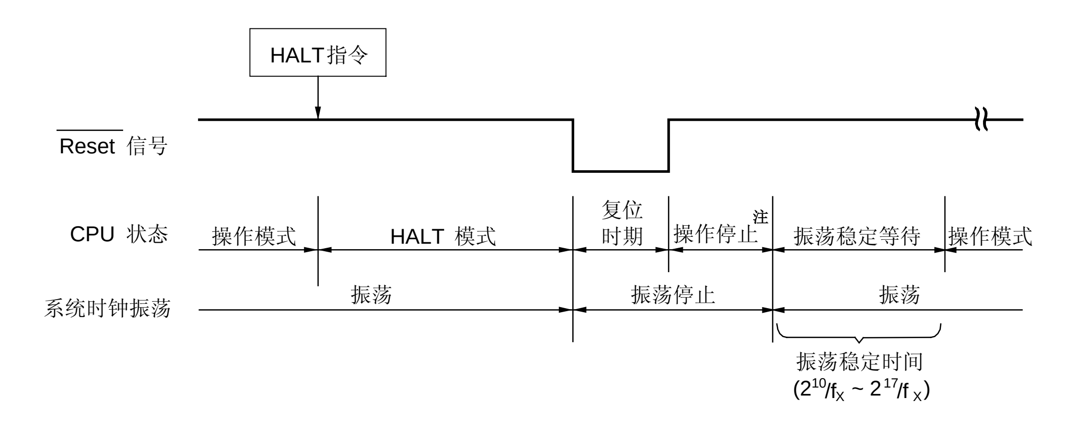
图 14-3. 由复位信号释放 HALT 模式

(1) 当 CPU 时钟为高速 **Ring-OSC** 时钟或外部输入时钟时



注 操作停止(277μs(最小),544μs(典型值),1.075ms(最大)) 参见可选字节。

(2) 当 CPU 时钟为晶体/陶瓷振荡时钟



注 操作停止(276μs(最小),544μs(典型值),1.074 ms(最大))参见可选字节。

备注 fx: 系统时钟振荡频率

表 14-3. 在 HALT 模式中与中断请求对应的操作

释放源	MK××	IE	操作
可屏蔽中断请求	0	0	执行下一个地址的指令
	0	1	执行中断服务程序
	1	×	保持 HALT 模式
复位信号产生	–	×	复位处理

×: 不必考虑

14.2.2 STOP 模式

(1) STOP 模式设置及操作状态

执行 STOP 指令设置 STOP 模式。

注意事项　由于中断请求信号用于释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。因此，在 STOP 模式中，执行 STOP 指令后恢复正常操作模式，然后操作停止 34 μs (典型值) (在经过由振荡稳定时间选择寄存器(OSTS)设置的等待时间后，晶体/陶瓷振荡器启动)。

STOP 模式中的操作状态如下所示。

表 14-4.　STOP 模式中的操作状态

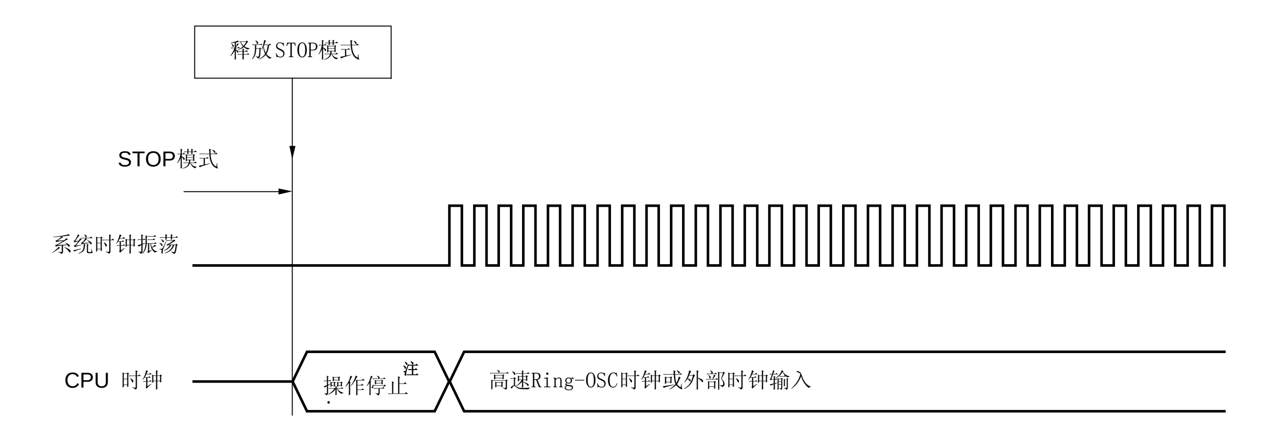
<table>
<tr><th colspan="2" rowspan="2">STOP 模式设置
项目</th><th rowspan="2">低速 Ring-OSC 振荡器不能被停止[注].</th><th colspan="2">低速 Ring-OSC 振荡器可被停止[注]</th></tr>
<tr><th>低速 Ring-OSC 振荡继续</th><th>低速 Ring-OSC 振荡停止</th></tr>
<tr><td colspan="2">系统时钟</td><td colspan="3">振荡停止</td></tr>
<tr><td colspan="2">CPU</td><td colspan="3">终止操作.</td></tr>
<tr><td colspan="2">端口(锁存)</td><td colspan="3">保持设置 STOP 模式前的状态</td></tr>
<tr><td colspan="2">16 位定时器/事件计数器 00</td><td colspan="3">终止操作</td></tr>
<tr><td colspan="2">8 位定时器 80</td><td colspan="3">终止操作</td></tr>
<tr><td rowspan="2">8 位定时器 H1</td><td>设置计数时钟为 f_{XP} 到 $f_{XP}/2^{12}$</td><td colspan="3">终止操作</td></tr>
<tr><td>设置计数时钟为 $f_{RL}/2^{7}$</td><td>可操作</td><td>可操作</td><td>终止操作</td></tr>
<tr><td rowspan="2">看门狗定时器</td><td>选择“外部硬件时钟”作为操作时钟</td><td>禁止设置</td><td colspan="2">终止操作</td></tr>
<tr><td>选择“低速 Ring-OSC 时钟”作为操作时钟</td><td>可操作(操作继续)</td><td colspan="2">终止操作.</td></tr>
<tr><td colspan="2">A/D 转换器</td><td colspan="3">终止操作</td></tr>
<tr><td colspan="2">串行接口 UART6</td><td colspan="3">终止操作</td></tr>
<tr><td colspan="2">POC 电路</td><td colspan="3">可操作</td></tr>
<tr><td colspan="2">低电压检测</td><td colspan="3">可操作</td></tr>
<tr><td colspan="2">外部中断</td><td colspan="3">可操作</td></tr>
</table>

注　可选字节可选择低速 Ring-OSC 振荡器“不能被停止”或“通过软件停止”(可选字节可参见 **第十八章　可选字节**)。

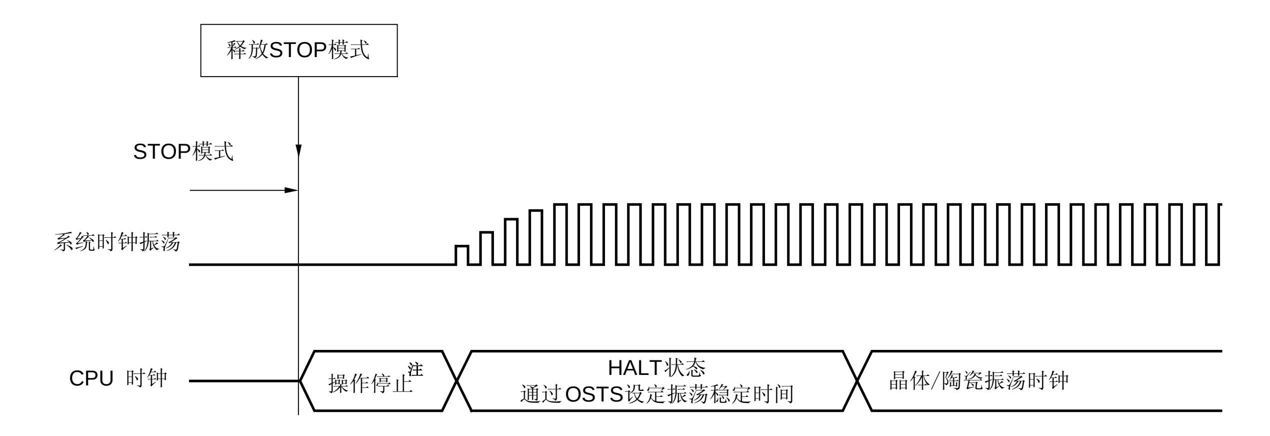
(2) 释放 STOP 模式

图 14-4. 当释放 STOP 模式时的操作时序

<1>如果选择高速 **Ring-OSC** 时钟或外部输入时钟作为系统时钟



<2> 如果选择晶体/陶瓷振荡时钟作为系统时钟



注 操作停止时间为 17μs(最小),34μs(典型值),和 67μs(最大)。

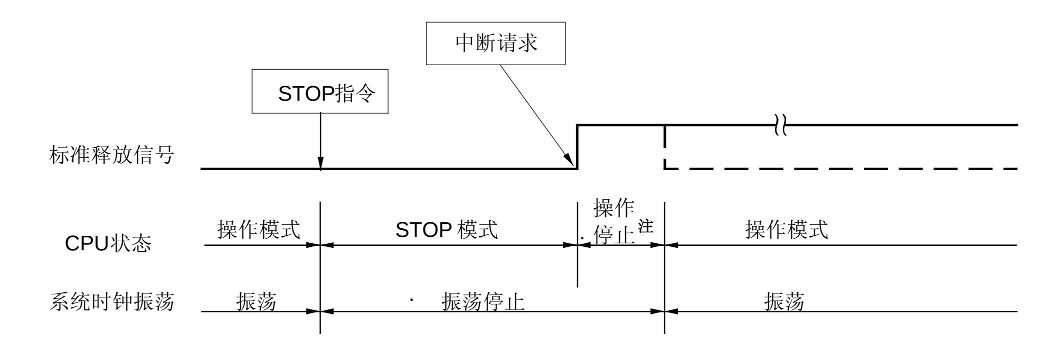
可由以下两种中断源释放 STOP 模式。

(a) 由未屏蔽的中断请求释放

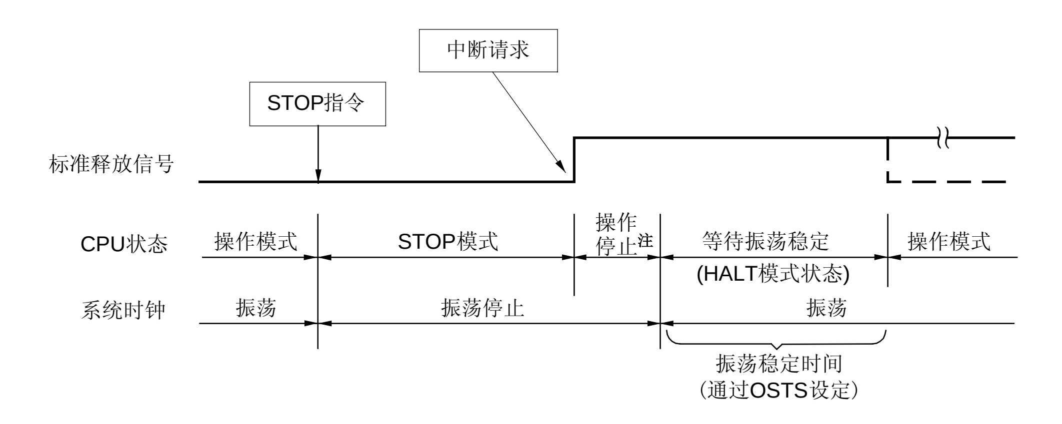
当产生未屏蔽的中断请求时，释放 STOP 模式。经历振荡器稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 14-5.　由中断请求释放 STOP 模式

(1)　如果 CPU 时钟为高速 **Ring-OSC** 时钟或外部输入时钟



(2)　如果 CPU 时钟为晶体/陶瓷振荡时钟



注　操作停止时间为 17μs(最小),34μs(典型值),和 67μs(最大)。

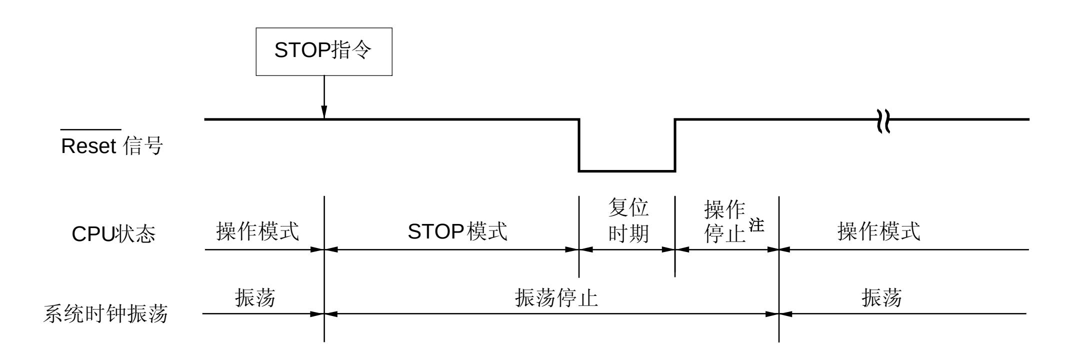
备注　虚线表示释放待机模式的中断请求被响应的情况。

(b) 由复位信号释放

当产生复位信号时，释放 STOP 模式，并在经过振荡稳定时间后执行复位操作。

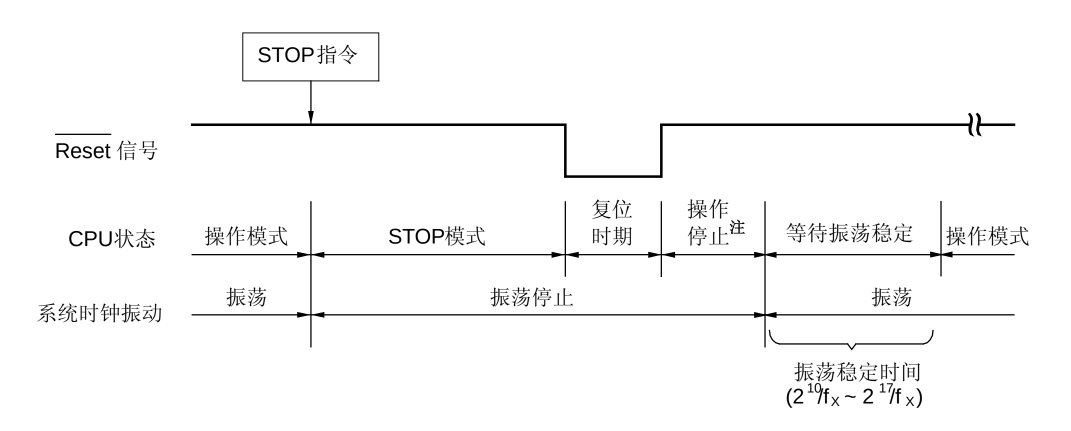
图 14-6. 由复位信号释放 STOP 模式

(1) 如果 CPU 时钟为高速 **Ring-OSC** 时钟或外部输入时钟



注 操作终止(277μs(最小),544μs(典型值),1.075ms(最大))参见可选字节。

(2) 如果 CPU 时钟为晶体/陶瓷振荡时钟



注 操作终止(276μs(最小),544μs(典型值),1.074ms(最大))参见可选字节。

备注 fx: 系统时钟振荡频率

表 14-5. 在 STOP 模式中与中断请求对应的操作

释放源	MK××	IE	操作
可屏蔽的中断请求	0	0	执行下一个地址的指令
	0	1	执行中断服务程序
	1	×	保持 STOP 模式
复位信号	–	×	进行复位

×: 不必考虑

第十五章　复位功能

以下 4 种操作可产生复位信号。

(1) 由 RESET 引脚输入的外部复位信号。
(2) 由看门狗定时器程序循环检测引起的内部复位。
(3) 通过比较上电清零（POC）电路的检测电压和供电电压引起的内部复位。
(4) 通过比较低电压检测器(LVI)的检测电压和供电电压引起的内部复位。

外部复位与内部复位在功能上没有什么区别。在这两种情况中，当输入复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 RESET 引脚输入为低电平或看门狗定时器溢出或 POC 和 LVI 电路的电压检测都可以引起复位，硬件的每项设置状态如表 15-1 所示。在复位输入期间或在复位释放后振荡器稳定时间内，每个引脚均为高阻抗，P130(低电平输出)除外。

当 RESET 引脚输入高电平时，复位释放，参考可选字节之后使用 CPU 时钟开始执行程序 (如果选择了晶体/陶瓷振荡器，在参考可选字节经过的时钟振荡稳定时间之后)。由看门狗定时器和时钟监视器引起的复位在复位后自动释放，并且 CPU 开始执行程序 (如果选择了晶体/陶瓷振荡器，在参考选项字节和经过时钟振荡稳定时间之后)。(参见**图 15-2** 到 **15-4**)。 由 POC 和 LVI 电路供电检测引起的复位，在复位后，当 VDD > VPOC 或 VDD > VLVI 时自动释放，并且 CPU 开始执行程序 (如果选择了晶体/陶瓷振荡器，在参考选项字节和经过时钟振荡稳定时间之后)。 (参见 **第十六章 POC 电路** 与 **第十七章　低电压检测**)。

注意事项
1. **对于外部复位，输入 RESET 引脚的低电平的时间应为 2μs 或更长。**
2. **复位输入期间，系统时钟和低速内部振荡时钟停止振荡。**
3. **RESET 引脚只能被复用输入端口引脚(P34)，如果在由 POC 电路释放复位后且在再次参考选项字节之前低电平输入到 RESET 引脚，78K0S/KU1+ 和 78K0S/KY1+复位。复位状态保持直到高电平输入到 RESET 引脚。**

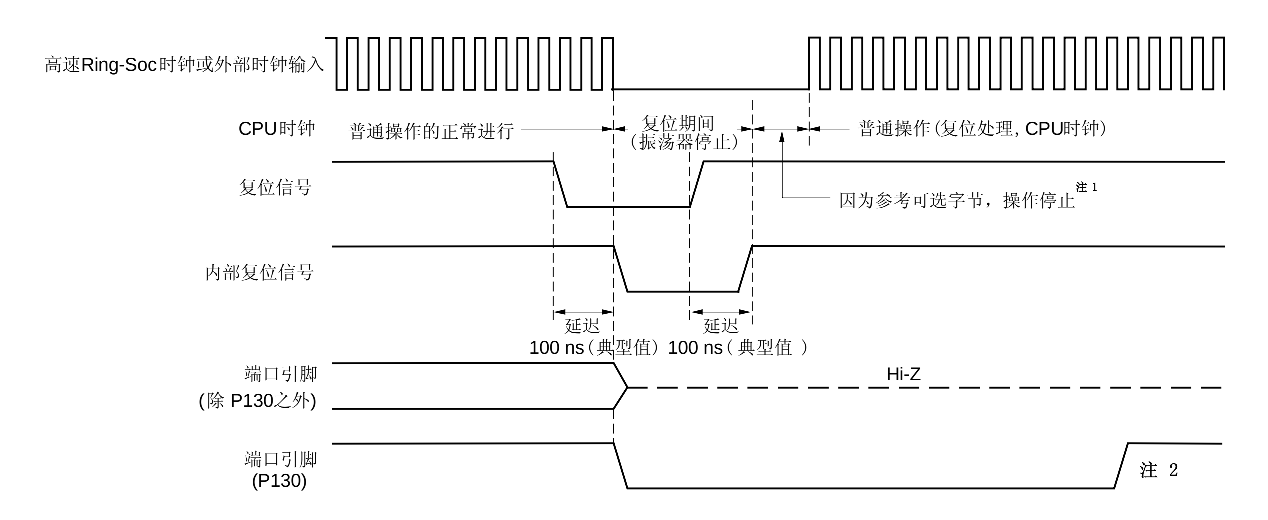
图 15-1.　复位功能的框图

注意事项 LVI 电路的内部复位信号不会使 LVI 电路复位。

备注
1. LVIM: 低电压检测寄存器
2. LVIS: 低电压检测电平选择寄存器

图 15-2. 由 $\overline{\text{RESET}}$ 输入引起的复位时序

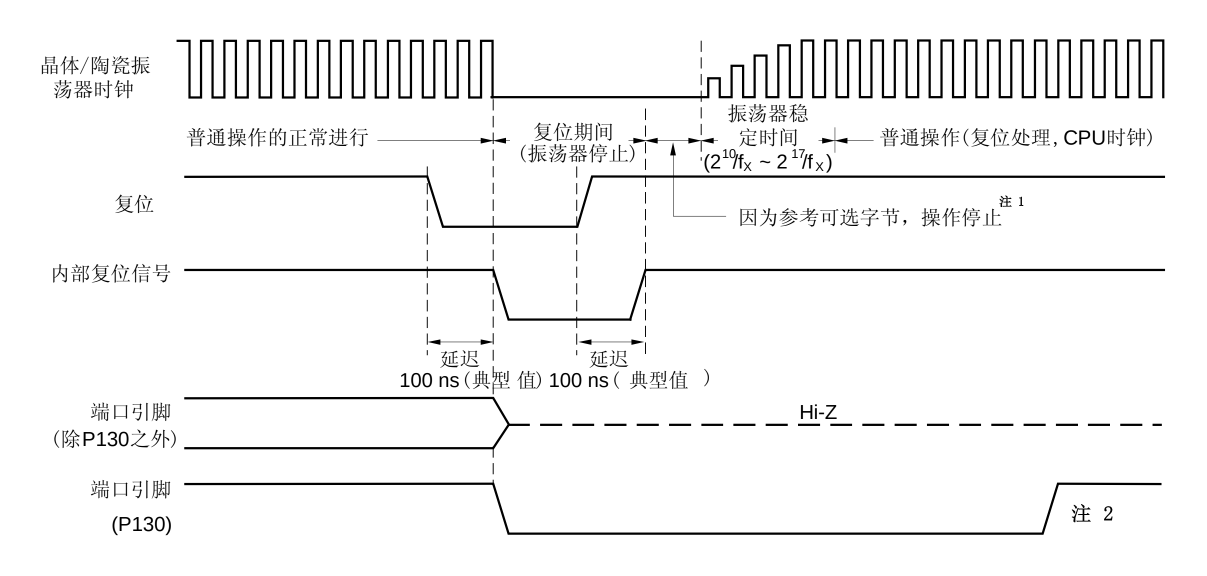
<1>高速 **Ring-OSC** 时钟或外部时钟输入



注 1. 操作停止时间为 277μs(最小), 544μs(典型值), 和 1.075ms(最大)。

2. 当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

<2>晶体/陶瓷振荡器时钟



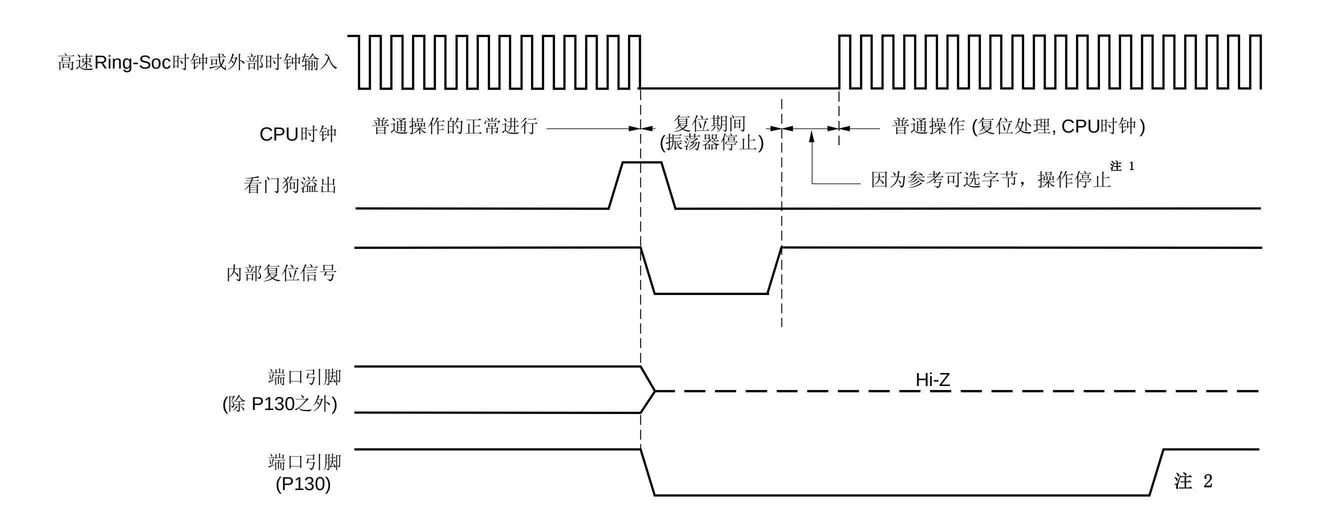
注 1. 操作停止时间为 276 μs (最小), 544 μs (典型值), 和 1.074 ms (最大)。

2. 当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

备注 f_X: 系统时钟振荡频率

图 15-3. 由看门狗定时器溢出引起的复位时序

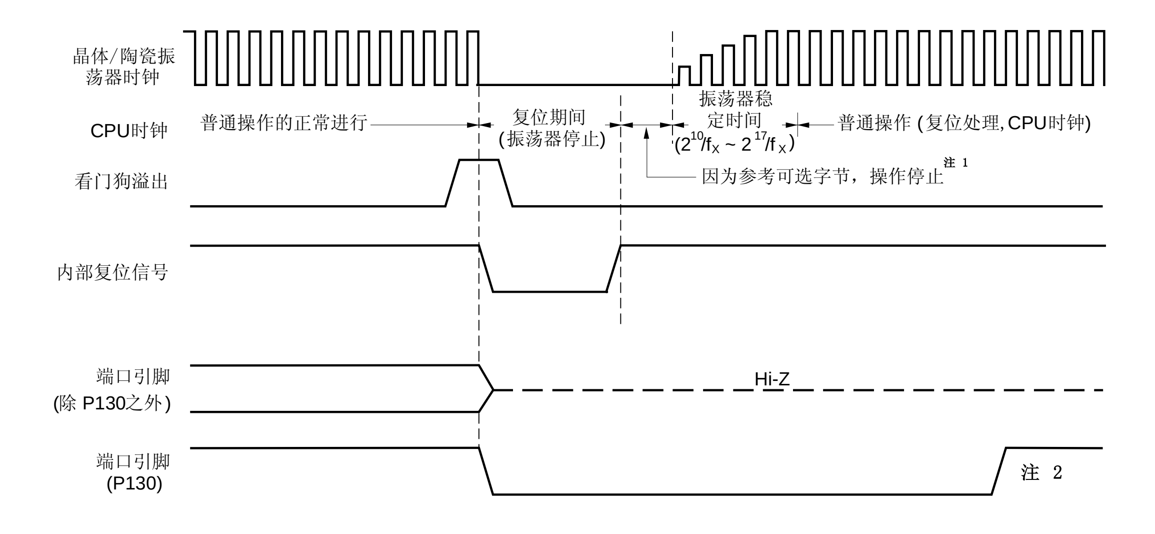
<1>高速 **Ring-OSC** 时钟或外部时钟输入



注 1. 操作停止时间为 277 μs (最小), 544 μs (典型值), 和 1.075 ms (最大)。
2. 当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

注意事项 在看门狗定时器内部复位的情况下，看门狗定时器也复位。

<2>晶体/陶瓷振荡器时钟



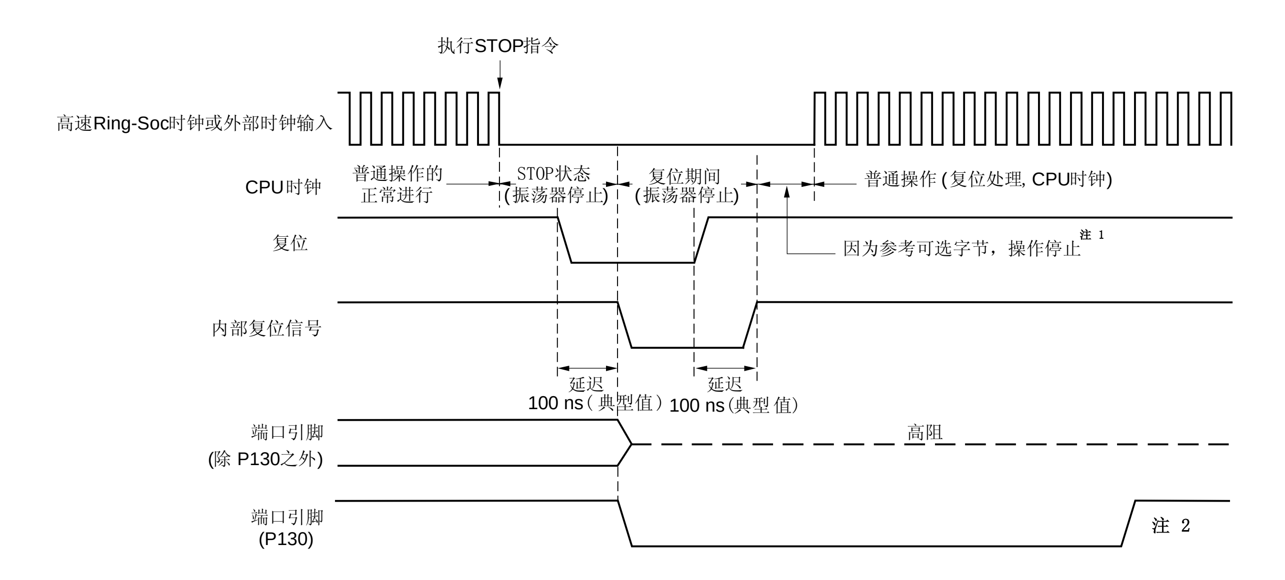
注 1. 操作停止时间为 276 μs (最小), 544 μs (典型值), 和 1.074 ms (最大)。
2. 当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

注意事项 在看门狗定时器内部复位的情况下，看门狗定时器也复位。

备注 fx: 系统时钟振荡频率

图 15-4.　在 STOP 模式中由 RESET 输入引起的复位时序

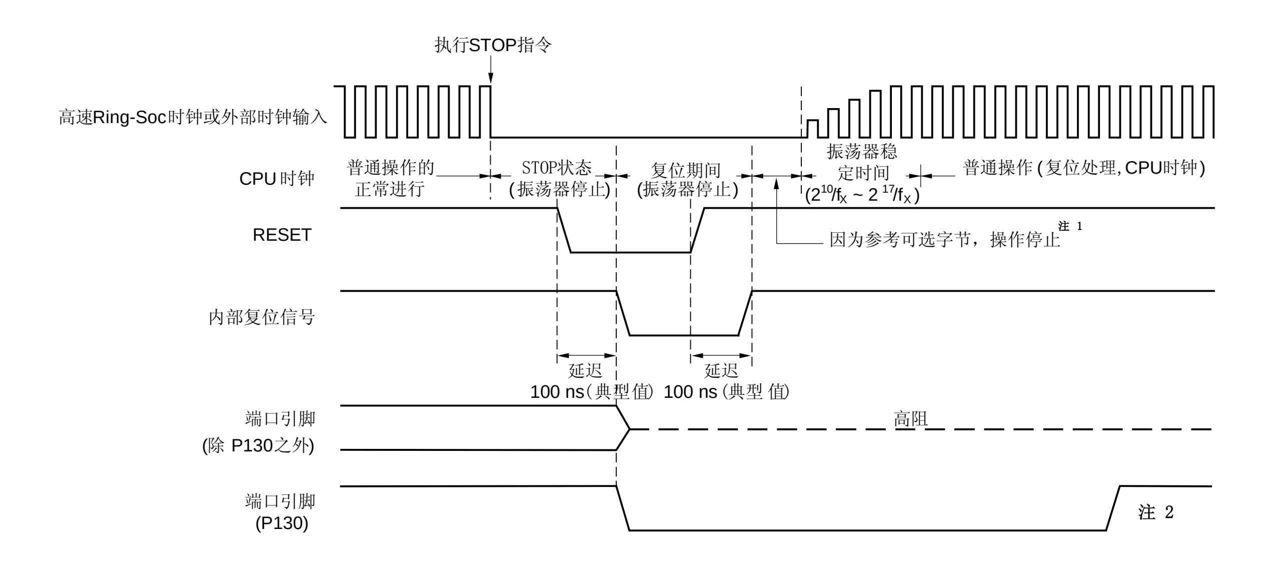
<1>高速 **Ring-OSC** 时钟或外部时钟输入



注　1.　操作停止时间为 277 μs（最小），544 μs（典型值），和 1.075 ms（最大）。

2.　当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

<2>晶体/陶瓷振荡器时钟



注　1.　操作停止时间为 276 μs（最小），544 μs（典型值），和 1.074 ms（最大）。

2.　当复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出高电平，P130 的输出信号就象 CPU 的复位信号被虚拟输出一样。

备注　1. POC 电路和低电压检测的复位时序，参见**第十六章 POC 电路**和**第十七章 低电压检测。**

2.　fx：系统时钟振荡频率

表 15-1. 复位响应后的硬件状态(1/2)

硬件		复位后的状态
程序计数器 (PC) 注 1		设置复位向量表的内容 (0000H 和 0001H)
堆栈指针(SP)		不确定
程序状态字 (PSW)		02H
RAM	数据存储器	不确定注 2
	通用目的寄存器	不确定注 2
端口 (P0,P2～P4,P12,P13)(输出锁存)		00H
端口模式寄存器 (PM0,PM2～PM4,PM12)		FFH
端口模式控制寄存器 (PMC2)		00H
上拉电阻选择寄存器 (PU0，PU2，PU3，PU4，PU12)		00H
处理器时钟控制寄存器(PCC)		02H
预处理器时钟控制寄存器 (PPCC)		02H
低速 Ring-OSC 模式寄存器 (LSRCM)		00H
振荡稳定时间选择寄存器 (OSTS)		不确定
16 位定时器 00	定时器计数器 00 (TM00)	0000H
	捕捉/比较寄存器 000, 010 (CR000, CR010)	0000H
	模式控制寄存器 00 (TMC00)	00H
	预分频器模式寄存器 00 (PRM00)	00H
	捕捉/比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H
8 位定时器 80	定时器计数器 80 (TM80)	00H
	比较寄存器(CR80)	不确定
	模式控制寄存器 80 (TMC80)	00H
8 位定时器 H1	比较寄存器(CMP01, CMP11)	00H
	模式寄存器 1 (TMHMD1)	00H
看门狗定时器	模式寄存器(WDTM)	67H
	使能寄存器(WDTE)	9AH
A/D 转换器	转换结果寄存器(ADCR, ADCRH)	不确定
	模式寄存器 (ADM)	00H
	模拟输入通道选择寄存器 (ADS)	00H

注　1.　当产生复位信号和经过振荡稳定时间时，只有 PC 的内容是不确定的。其他硬件单元的状态保持不变。

　　2.　在待机模式中，保持复位后的状态。

表 15-1.　复位响应后的硬件状态(2/2)

硬件		复位后的状态
串行接口 UART6	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口操作模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送错误状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率发生控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入选择控制寄存器(ISC)	00H
乘法器	16 位乘法结果存储寄存器 (MUL0)	不确定
	数据寄存器 (MRA0, MRB0)	不确定
	控制寄存器　(MULC0)	00H
复位功能	复位控制标志寄存器(RESF)	00H 注
低电压检测	低电压检测寄存器(LVIM)	00H 注
	低电压检测电平选择寄存器(LVIS)	00H 注
中断	中断请求标志寄存器(IF0, IF1)	00H
	中断屏蔽标志寄存器(MK0, MK1)	FFH
	外部中断模式寄存器(INTM0, INTM1)	00H
Flash 存储器	Flash 保护命令寄存器(PFCMD)	不确定
	Flash 状态寄存器(PFS)	00H
	Flash 编程模式控制寄存器(FLPMC)	不确定
	Flash 编程命令寄存器(FLCMD)	00H
	Flash 地址指针 L (FLAPL)	不确定
	Flash 地址指针 H (FLAPH)	
	Flash 地址指针 H 比较寄存器 (FLAPHC)	00H
	Flash 地址指针 L 比较寄存器 (FLAPLC)	00H
	Flash 写缓冲寄存器 (FLW)	00H

注　　这些值依据复位源的不同而不同如下。

复位源 / 寄存器	$\overline{\text{RESET}}$ 输入	由 POC 复位	由 WDT 复位	由 LVI 复位
RESF	见 表 15-2.			
LVIM	清零(00H)	清零(00H)	清零(00H)	保持
LVIS				

15.1 确认复位源的寄存器

78K0S/KB1+中有许多内部复位源。复位控制标志寄存器(RESF)用于存储产生复位请求的复位源。
可由 8 位存储器操作指令读取 RESF。
$\overline{\text{RESET}}$ 输入、由 POC 电路引起的复位输入以及读取 RESF 可将 RESF 清零。

图 15-5. 复位控制标志寄存器(RESF)的格式

地址： FF54H 复位后： 00H[注] R

符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	由看门狗定时器产生的内部复位请求(WDT)
0	不产生内部复位请求，或对 RESF 清零.
1	产生内部复位请求

LVIRF	由低电压检测器产生的内部复位请求(LVI)
0	不产生内部复位请求，或对 RESF 清零
1	产生内部复位请求

注 复位后的值依据复位源而不同。

注意事项 不能通过 1 位存储器操作指令读取数据。

产生复位请求时的 RESF 状态如表 15-2 所示。

表 15-2. 产生复位请求时的 RESF 状态

标志 \ 复位源	$\overline{\text{RESET}}$ 输入	由 POC 复位	由 WDT 复位	由 LVI 复位
WDTRF	清零(0)	清零(0)	设置 (1)	保持
LVIRF			保持	设置 (1)

第十六章　上电清零电路

16.1 上电清零电路的功能

上电清零电路(POC) 有如下功能。

- 在上电时产生内部复位信号
- 比较电源电压(V_{DD}) 和检测电压 (V_{POC} = 2.1 V ±0.1 V), 并在 V_{DD} < V_{POC} 时产生内部复位信号。
- 比较电源电压(V_{DD}) 和检测电压 (V_{POC} = 2.1 V ±0.1 V), 并在 V_{DD} ≥ V_{POC} 时释放内部复位信号。

注意事项　1. 如果 POC 电路产生内部复位信号，则将复位标志寄存器（RESF）清零。

2. POC 电路的检测电压(V_{POC})值范围为 2.1 V ±0.1 V, 故正常工作电压范围为 2.2 ～5.5 V。

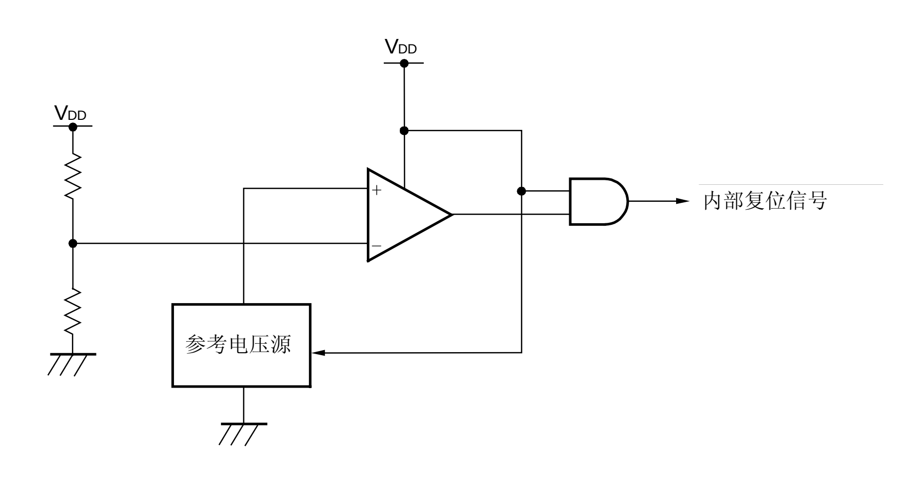
备注　本产品集成了多种用以产生内部复位信号的硬件功能。当看门狗定时器（WDT）和低电压检测电路（LVI）引起复位时，RESF 中有一个标志用于指示复位原因。当 WDT 和 LVI 产生内部复位信号时,RESF 不能被清零,且标志位被置 1。

如需了解 RESF 的详细内容, 参见 **第十五章 复位功能。**

16.2 上电清零电路的配置

POC 电路的框图如图 16-1 所示。

图 16-1. POC 电路的框图



16.3 上电清零电路的操作

在 POC 电路中比较电源电压(V_{DD}) 和检测电压 (V_{POC} = 2.1 V ±0.1 V),当 $V_{DD} < V_{POC}$ 时, 产生内部复位信号,当 $V_{DD} \geq V_{POC}$ 时,释放内部复位信号。

图 16-2. POC 电路中产生内部复位信号的时序

电源电压(V_{DD})

POC检测电压

(V_{POC}= 2.1V±0.1V)

时间

内部复位信号

备注　内部复位信号为低电平有效。

16.4 上电清零电路的使用注意事项

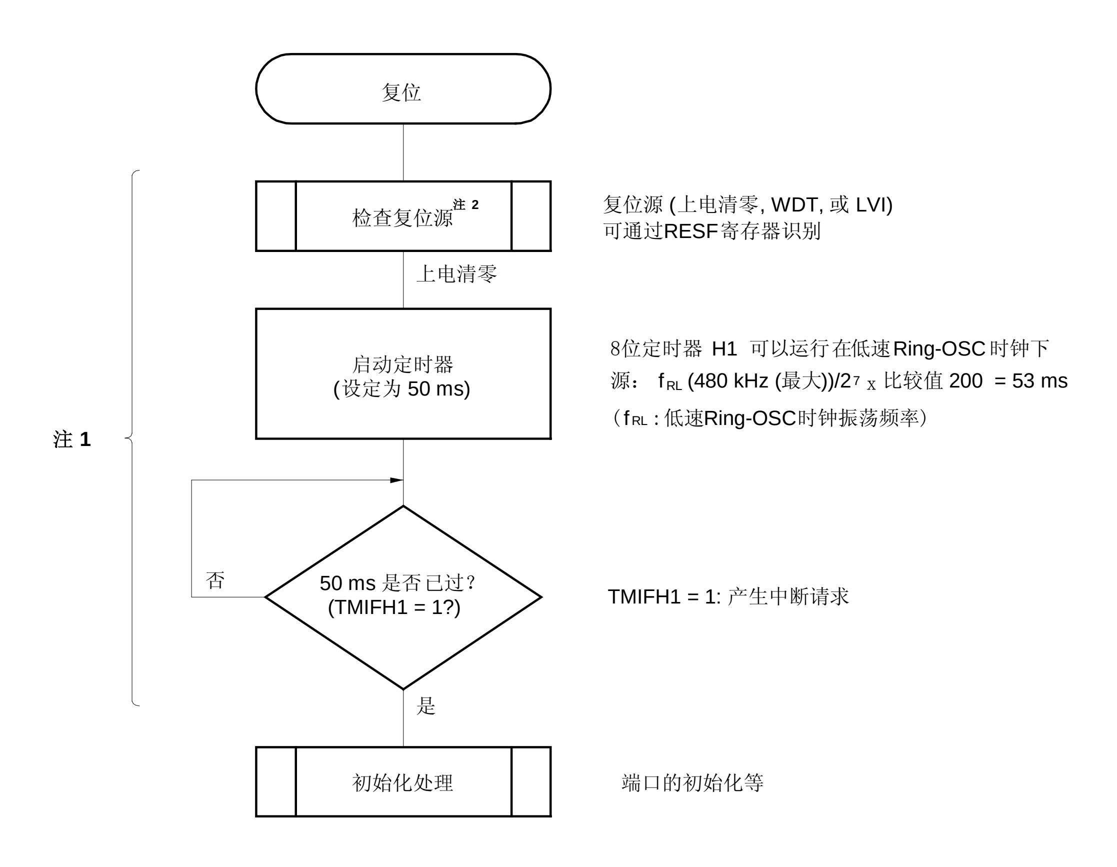
对于一个电源电压在某段时间内在 POC 检测电压(VPOC)附近波动的系统，可能会反复进行复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所经历的时间。

<方法>

在释放复位信号后，通过软件计数器（使用一个定时器）等待系统电源电压的波动期，然后对端口初始化。

图 16-3. 复位释放后软件处理过程示例(1/2)

• 如果在电源电压接近 POC 检测电压时，其波动时间小于等于 50ms

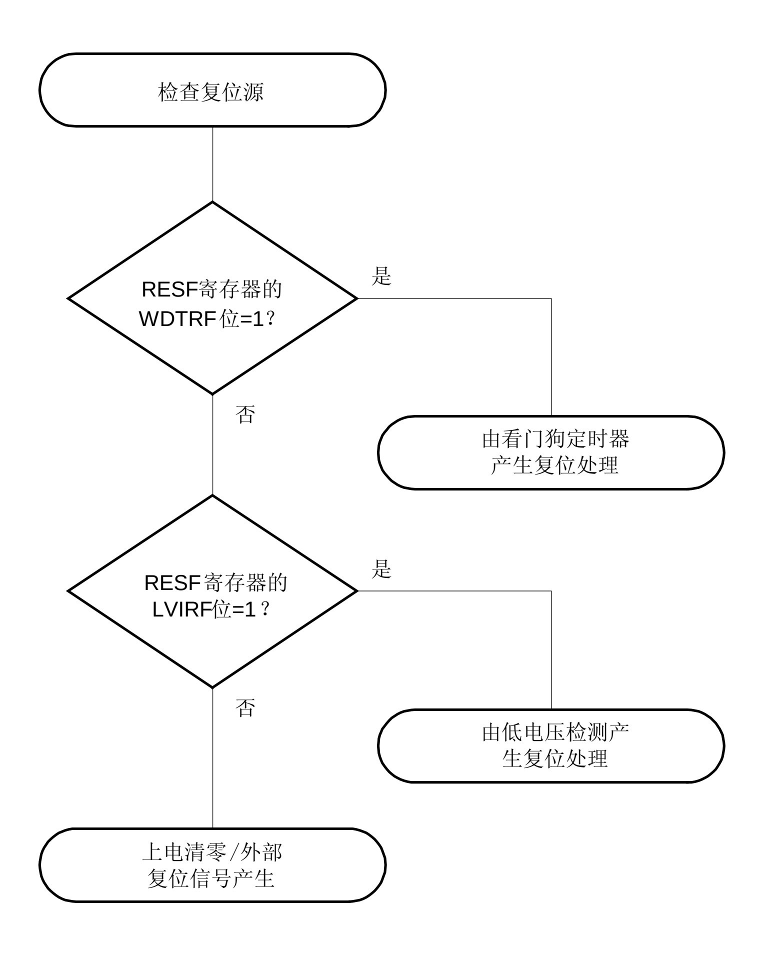


注　**1.** 如果在此期间再次产生复位，初始化操作不能开始。

　　2. 流程图显示在下页。

图 16-3. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



第十七章　低电压检测器

17.1　低电压检测器的功能

低电压检测器 (LVI) 有以下功能。

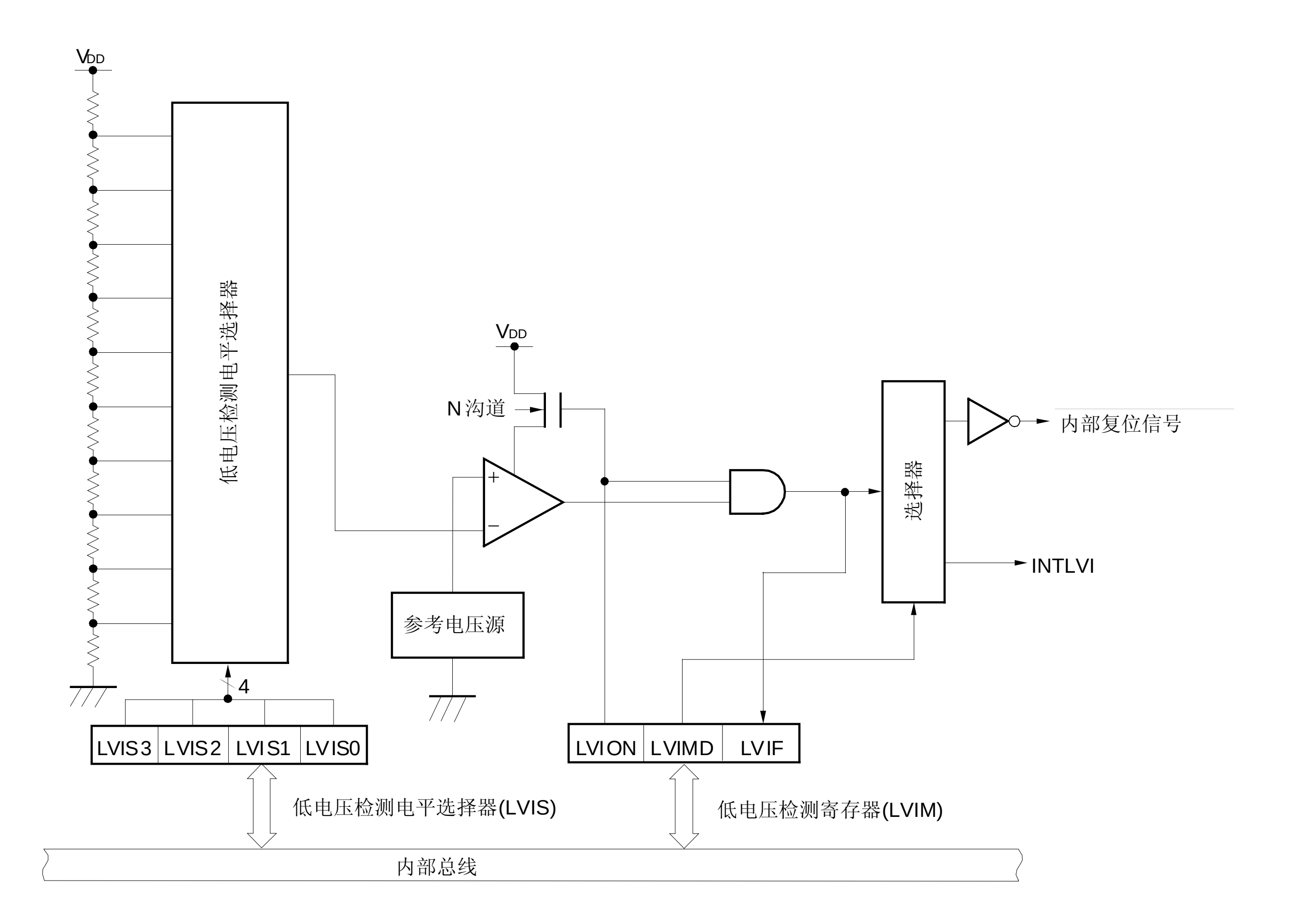
- 比较电源电压 (V_{DD})和检测电压 (V_{LVI})，当 V_{DD} < V_{LVI} 时，产生内部中断信号或内部复位信号
- 可由软件改变电源电压的检测电平 (十个电平)
- 可由软件选择中断或复位功能
- 在 STOP 模式下可操作

在使用低电压检测器进行复位时，如果产生复位，则复位控制标志寄存器(RESF)的第 0 位(LVIRF) 被设置为 1。需要了解 RESF 的详细内容，可参见 **第十五章 复位功能**。

17.2　低电压检测器的配置

低电压检测器的框图如图 17-1 所示。

图 17-1.　低电压检测器的框图



17.3 控制低电压检测器的寄存器

低电压检测器由以下寄存器控制

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)

(1) 低电压检测寄存器(LVIM)

该寄存器用于设置低电压检测和操作模式。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位输入将 LVIM 清零[注 1]。

图 17-2. 低电压检测寄存器(LVIM)的格式

地址:FF50H 复位后:00H [注 1] 读/写 [注 2]

符号	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION [注 3]	使能低电压检测操作
0	禁止操作
1	允许操作

LVIMD	低电压检测操作模式的选择
0	当电源电压 (V_{DD}) < 检测电压 (V_{LVI})时产生中断信号
1	当电源电压 (V_{DD}) < 检测电压 (V_{LVI})时产生内部复位信号

LVIF [注 4]	低电压检测标志
0	当电源电压(V_{DD}) ≥检测电压(V_{LVI}), 或当禁止操作时
1	电源电压 (V_{DD}) <检测电压(V_{LVI})

注 1. 对于由 LVI 产生的一个复位, LVIM 的值未被初始化。
2. 第 0 位只读。
3. 当 LVION =1 时, LVI 电路中的比较器开始运行。从设置 LVION=1 到 LVIF=1,至少经过 0.2 ms。
4. 当 LVION = 1 且 LVIMD = 0 时, LVIF 的值作为中断请求信号 INTLVI 输出。

注意事项 1. 可在执行完以下任一过程后终止 LVI。
- 当使用 8 位存储器操作指令时: 写 00H 到 LVIM。
- 当使用 1 位存储器操作指令时: 对 LVION 清零。

2. 第 2～6 位必须清零。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器用于选择低电压检测电平

可由 8 位存储器操作指令设置该寄存器

复位输入将 LVIS 清零[注]。

图 17-3. 低电压检测电平选择寄存器 (LVIS)的格式

地址： FF51H 复位后： 00H [注] R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V_{LVI0} (4.3 V ±0.2 V)
0	0	0	1	V_{LVI1} (4.1 V ±0.2 V)
0	0	1	0	V_{LVI2} (3.9 V ±0.2 V)
0	0	1	1	V_{LVI3} (3.7 V ±0.2 V)
0	1	0	0	V_{LVI4} (3.5 V ±0.2 V)
0	1	0	1	V_{LVI5} (3.3 V ±0.15 V)
0	1	1	0	V_{LVI6} (3.1 V ±0.15 V)
0	1	1	1	V_{LVI7} (2.85 V ±0.15 V)
1	0	0	0	V_{LVI8} (2.6 V ±0.1 V)
1	0	0	1	V_{LVI9} (2.35 V ±0.1 V)
其它设置				禁止设置

注 对于由 LVI 产生的一个复位，LVIM 的值未被初始化。

注意事项 第 4 至 7 位必须清零。

17.4 低电压检测器的操作

低电压检测器可用在以下两种模式中：

- 用作复位
  比较电源电压 (V_{DD})和检测电压(V_{LVI})，当 V_{DD} < V_{LVI} 产生内部复位信号，当 V_{DD} ≥ V_{LVI} 释放内部复位信号。
- 用作中断
  比较电源电压 (V_{DD})和检测电压(V_{LVI})，当 V_{DD} < V_{LVI} 产生中断信号。

操作设置如下。

(1) 用作复位

- 启动操作时

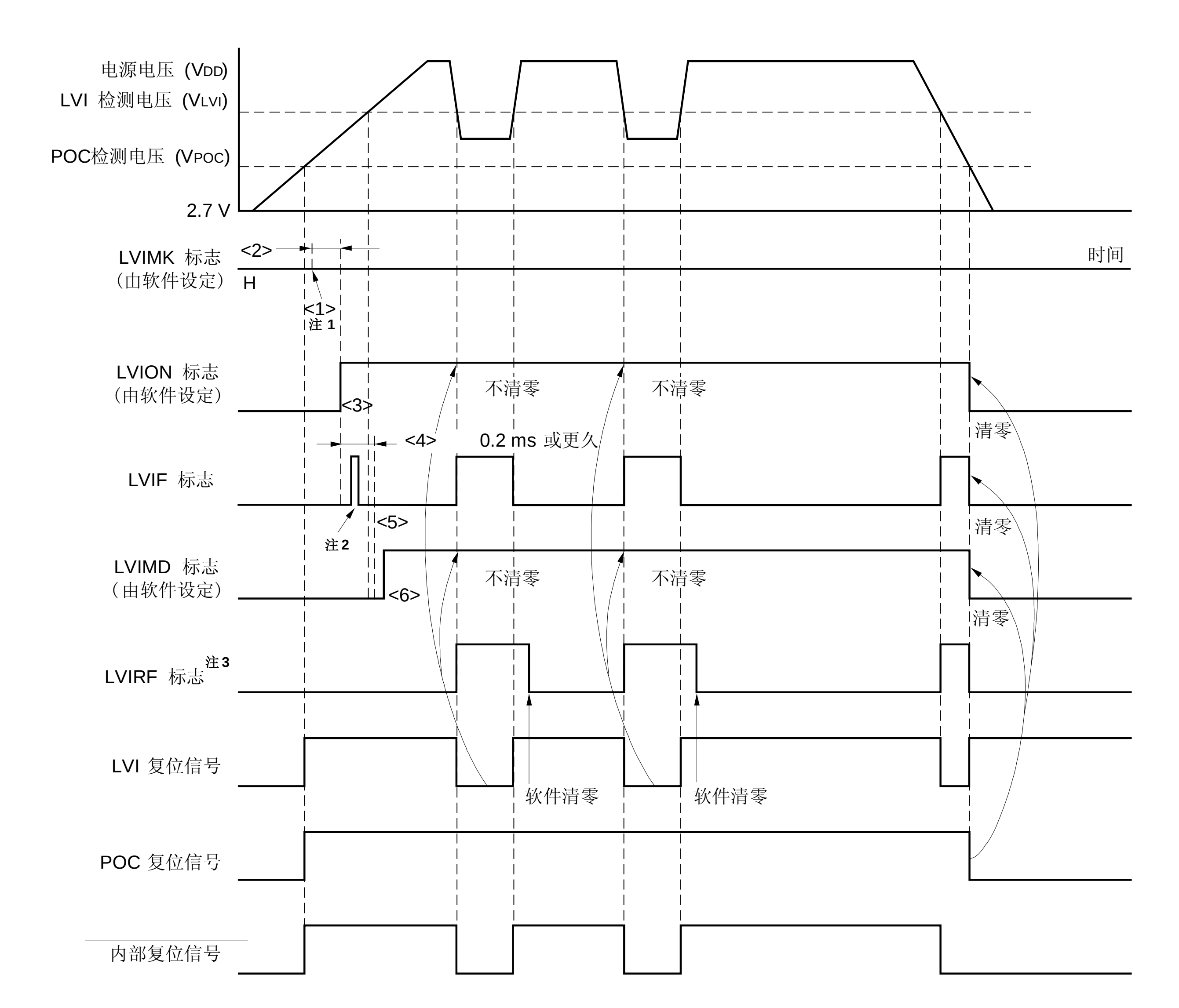
<1> 屏蔽 LVI 中断 (LVIMK = 1)。
<2> 使用低电压检测电平选择寄存器(LVIS)的第 3 ～0 位 (LVIS3 ～ LVIS0)设置检测电压。
<3> 设置低电压检测寄存器(LVIM)的第 7 位(LVION)＝1 (允许 LVI 操作)。
<4> 使用软件等待至少 0.2 ms。
<5> 当根据 LVIM 的第 0 位(LVIF)检查到 “电源电压(V_{DD}) ≥ 检测电压(V_{LVI})” 时，等待过程结束。
<6> 设置 LVIM 的第 1 位(LVIMD)为 1(当电源电压(V_{DD})<检测电压(V_{LVI})时产生内部复位信号)。

图 17-4 显示了由低电压检测器产生的内部复位信号的时序。该时序图中的数字与上面的<1>～<6>对应。

注意事项 1. 必须执行过程<1>，当 LVIMK = 0 时,在过程<3>之后可能立即会产生中断。
2. 当 LVIM 被设置为 1 时，如果电源电压 (V_{DD}) ≥ 检测电压 (V_{LVI})，则不会产生内部复位信号。

- 当终止操作时
  必须执行以下过程之一：
  - 当使用 8 位存储器操作指令时： 写 00H 到 LVIM。
  - 当使用 1 位存储器操作指令时： 依次将 LVIMD 和 LVION 清零。

图 17-4. 低电压检测器内部复位信号产生时序



注 1. 由复位输入将 LVIMK 标志设置为 "1"。

2. 可以设置 LVIF 标志为"1"。

3. LVIRF 是复位控制标志寄存器(RESF)的第 0 位。需要了解 RESF 的详细内容,可参见 **第十五章 复位功能。**

备注 图 17-4 中的<1>至<6> 与 17.4 (1) **用作复位**中"当启动操作时"描述部分的<1>至<6>对应。

(2) 用作中断

- 当启动操作时

〈1〉 屏蔽 LVI 中断(LVIMK = 1)。

〈2〉 使用低电压检测电平选择寄存器(LVIS)的第 3～0 位(LVIS3～LVIS0)设置检测电压。

〈3〉 设置低电压检测寄存器(LVIM)的第 7 位 (LVION)＝1(允许 LVI 操作)。

〈4〉 使用软件等待至少 0.2 ms 的时间。

〈5〉 当根据 LVIM 的第 0 位(LVIF)检查到 "电源电压(V_{DD}) ≥ 检测电压(V_{LVI})" 时，等待过程结束。

〈6〉 将 LVI 的中断请求标志(LVIIF)清零。

〈7〉 释放 LVI 的中断屏蔽标志 (LVIMK)。

〈8〉 执行 EI 指令(当使用中断向量时)。

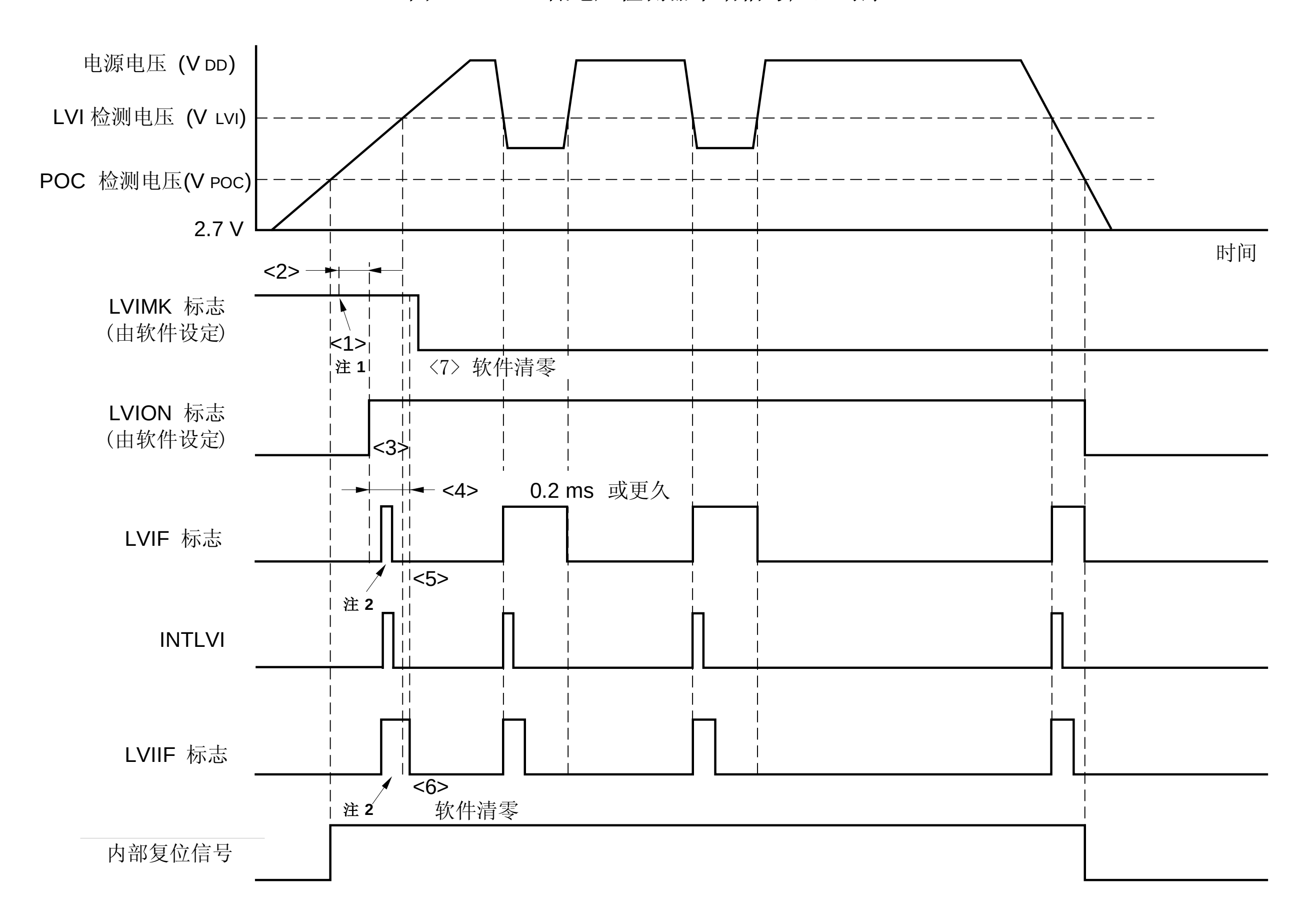
图 17-5 显示了由低电压检测器产生的中断信号的时序，该时序图中的数字与上面的〈1〉至〈7〉对应。

- 当终止操作时

  必须执行以下过程之一。

  - 当使用 8 位存储器操作指令时： 写 00H 到 LVIM。
  - 当使用 1 位存储器操作指令时： 对 LVION 清零。

图 17-5. 低电压检测器中断信号产生时序



注 1. 由复位输入把 LVIMK 标志设置为"1"。

2. LVIF 和 LVIIF 标志可以设置为"1"。

备注 图 17-5 中的〈1〉至〈7〉与 17.4 (2) **作为中断使用**中"当启动操作时"描述部分的〈1〉至〈7〉对应。

17.5 低电压检测器的注意事项

在一个电源电压(VDD)在某段时间内在LVI检测电压附近波动的系统中，以下操作决定于如何使用低电压检测器。

〈1〉 用作复位

系统可能会反复进行复位和释放复位。

在这种情况下,可采用下面的方法(1)任意设置从复位释放到微控制器的启动所经历的时间。

〈2〉 用作中断

可能会频繁产生中断请求。可采用方法(2)。

在该系统中可采用如下方法。

〈方法〉

(1) 用作复位

在释放复位信号后，通过软件计数器（使用一个定时器）等待电源电压的波动期，然后对端口初始化(参见图 17-6)。

(2) 用作中断

执行低电压检测过程[注]，在 LVI 中断服务程序中，利用低电压检测寄存器(LVIM)的第 0 位(LVIF),检查“电源电压(V_{DD}) ≥ 检测电压(V_{LVI})”。

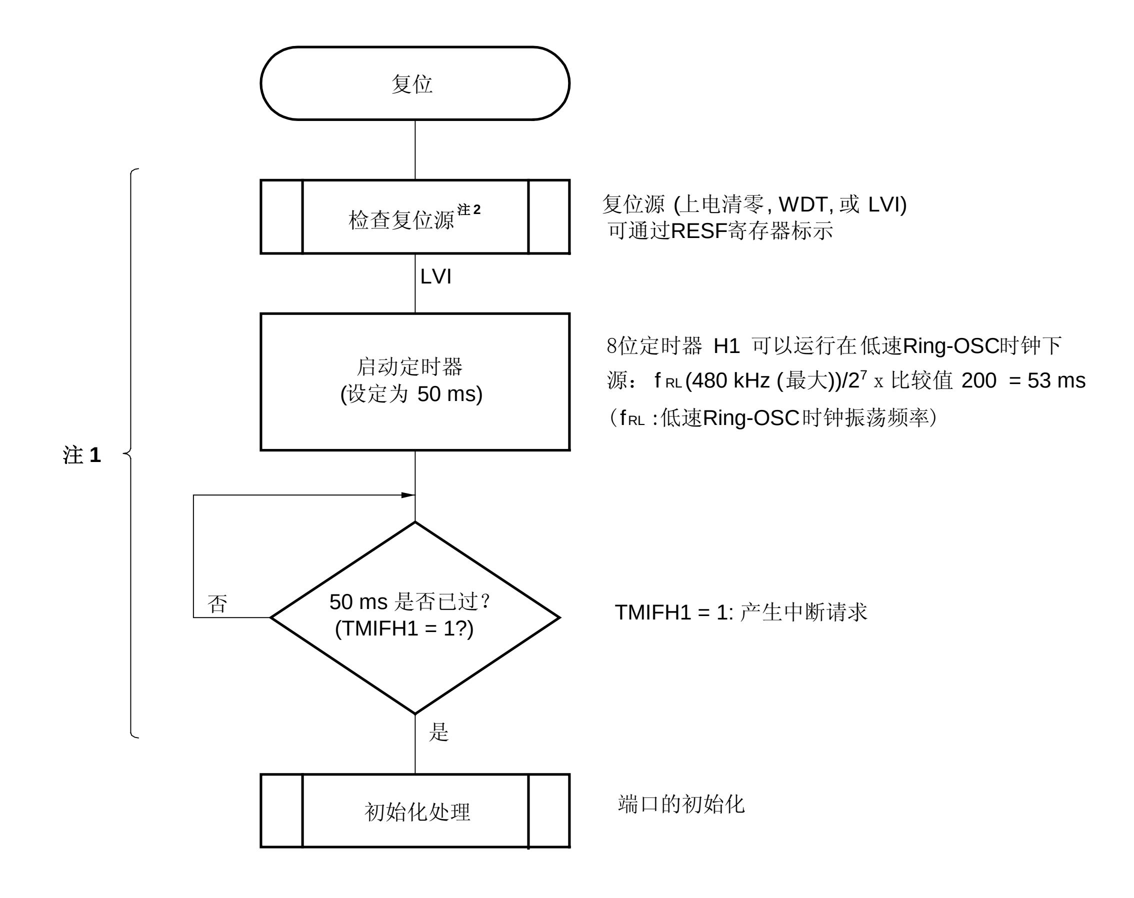
将中断请求标志寄存器(IF0)的第1位 (LVIIF)清零并允许中断（EI）。

在系统中当电源电压接近 LVI 检测电压,电源电压的波动时间很长时，则要进行等待,利用 LVIF 标志检查“电源电压 (V_{DD}) ≥ 检测电压(V_{LVI})”的情况,然后清LVIIF 为零，并允许中断（EI）。

注 对于低电压检测过程，CPU 时钟速度切换至低速并且 A/D 转换器停止等。

图 17-6. 复位释放后的软件处理过程示例(1/2)

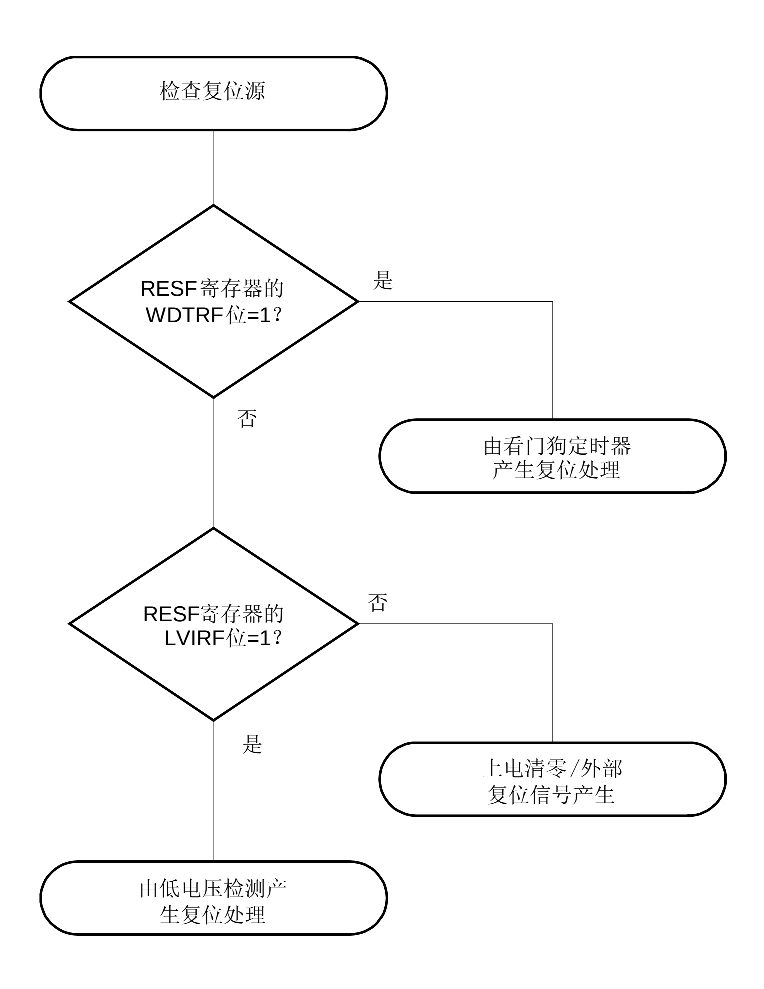
- 如果电源电压接近 LVI 检测电压,其波动时间小于等于 50ms。



注 1. 如果在这期间再次产生复位信号，则初始化处理不启动。
2. 流程图如下页所示。

图 17-6. 复位释放后的软件处理过程示例(2/2)

- 检查复位源



第十八章　可选字节

78K0S/KB1+有一个名为可选字节的区域，在 Flash 存储器的地址 0080H 处。使用该产品时，必须通过可选字节设置如下功能。

1. **系统时钟源选择**
   - 高速 Ring-OSC 时钟
   - 晶体/陶瓷振荡时钟
   - 外部时钟输入

2. **低速 Ring-OSC 时钟**
   - 不能被停止
   - 可通过软件停止

3. **$\overline{\text{RESET}}$ 引脚的控制**
   - 用作 $\overline{\text{RESET}}$ 引脚
   - $\overline{\text{RESET}}$ 引脚可被用作输入端口引脚(P34)

4. **上电时或复位释放后振荡稳定时间**
   - $2^{10}/f_X$
   - $2^{12}/f_X$
   - $2^{15}/f_X$
   - $2^{17}/f_X$

图 18-1.　可选字节的配置

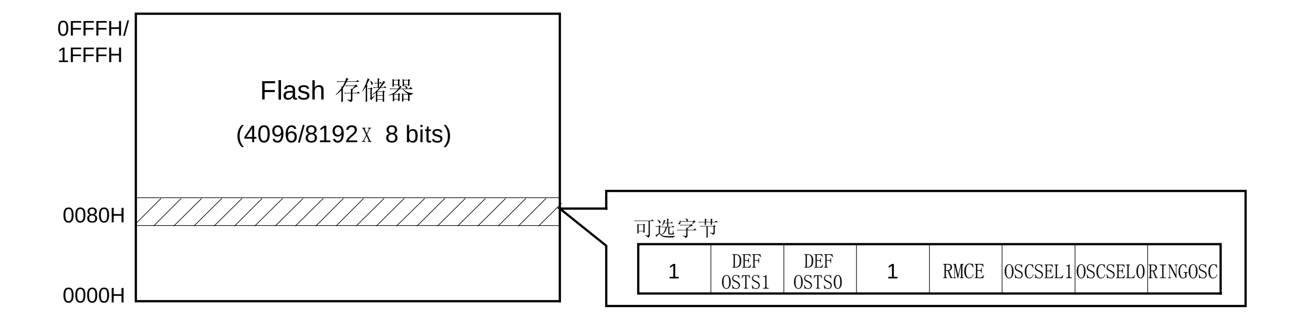


图 18-2.　可选字节的格式 (1/2)

地址：　0080H

7	6	5	4	3	2	1	0
1	DEFOSTS1	DEFOSTS0	1	RMCE	OSCSEL1	OSCSEL0	RINGOSC

RINGOSC	低速 Ring-OSC 时钟
1	不能停止(即使 LSRSTOP 位被写为 1，振荡也不会停止)
0	通过软件停止(当 LSRSTOP 位被写为 1 时，振荡停止)

注意事项 1.　如果选择低速 Ring-OSC 时钟振荡不能被停止，看门狗定时器(WDT)的计数器时钟就会固定为低速 Ring-OSC 时钟。

2.　如果选择低速 Ring-OSC 时钟可以通过软件停止，不管低速 Ring-OSC 模式寄存器(LSRCM)的第 0 位(LSRSTOP)设置如何，WDT 的计数器时钟在 HALT/STOP 模式下被停止,。同样地，当选择除低速 Ring-OSC 时钟之外的任何一个时钟作为 WDT 的计数器时钟时，时钟提供也被停止。但是，如果选择低速 Ring-OSC 时钟作为 8 位定时器 H1 的计数时钟，那么在低速 Ring-OSC 工作时(LSRSTOP = 0),在 STOP 模式下计数时钟可提供给 8 位定时器 H1 使用。

OSCSEL1	OSCSEL0	系统时钟源选择
0	0	晶体/陶瓷振荡时钟
0	1	外部时钟输入
1	×	高速 Ring-OSC 时钟

注意事项 因为 X1 和 X2 引脚也作为 P121 和 P122 引脚使用， X1 和 X2 引脚使用的条件不同，取决于系统时钟源的选择。

(1) 选择晶体/陶瓷振荡时钟
X1 和 X2 引脚不能作为 I/O 端口使用，因为它们正作为时钟输入端口

(2) 选择外部时钟输入
因为 X1 引脚作为外部时钟输入口，P121 口不能作为 I/O 口使用

(3) 选择高速内部振荡器时钟
P121 和 P122 端口可作为 I/O 端口

备注　　× ： 不考虑

RMCE	$\overline{\text{RESET}}$ 引脚的控制
1	$\overline{\text{RESET}}$ 引脚用作复位输入
0 [注]	$\overline{\text{RESET}}$ 引脚用作输入引脚(P34)

注　　当将 RMCE 清零时，需连接上拉电阻。

注意事项 在通过上电清零功能释放复位状态后并且在再次参考可选字节之前，一个低电平输入到 $\overline{\text{RESET}}$ 引脚，78K0S/KB1+复位，并且状态保持到一个高电平输入到 $\overline{\text{RESET}}$ 引脚。

图 18-2. 可选字节的格式 (2/2)

DEFOSTS1	DEFOSTS0	上电时或复位释放后振荡稳定时间
0	0	2^{10}/fx (102.4 μs)
0	1	2^{12}/fx (409.6 μs)
1	0	2^{15}/fx (3.27 ms)
1	1	2^{17}/fx (13.1 ms)

注意事项 仅当选晶体/陶瓷振荡时钟作为为系统时钟源时,这项设定才有效。 如果选用高速 Ring-OSC 时钟或外部时钟输入作为系统时钟源时,无需等待时间。

备注 1. ( ): fx = 10 MHz

2. 需要了解振荡器的振荡稳定时间，可参考所用的振荡器的特性。

设置可选字节的软件代码示例显示如下。

```
OPT CSEG AT 0080H
DB 10010101B            ; 设置可选字节
                        ; 停止低速 Ring-OSC
                        ; 选择高速 Ring-OSC 作为系统时钟源
                        ; 把 RESET 引脚作为输入端口引脚 (P34)使用
                        ; 振荡稳定时间为最小值 (2^10/fx)
```

第十九章　FLASH 存储器

19.1　特征

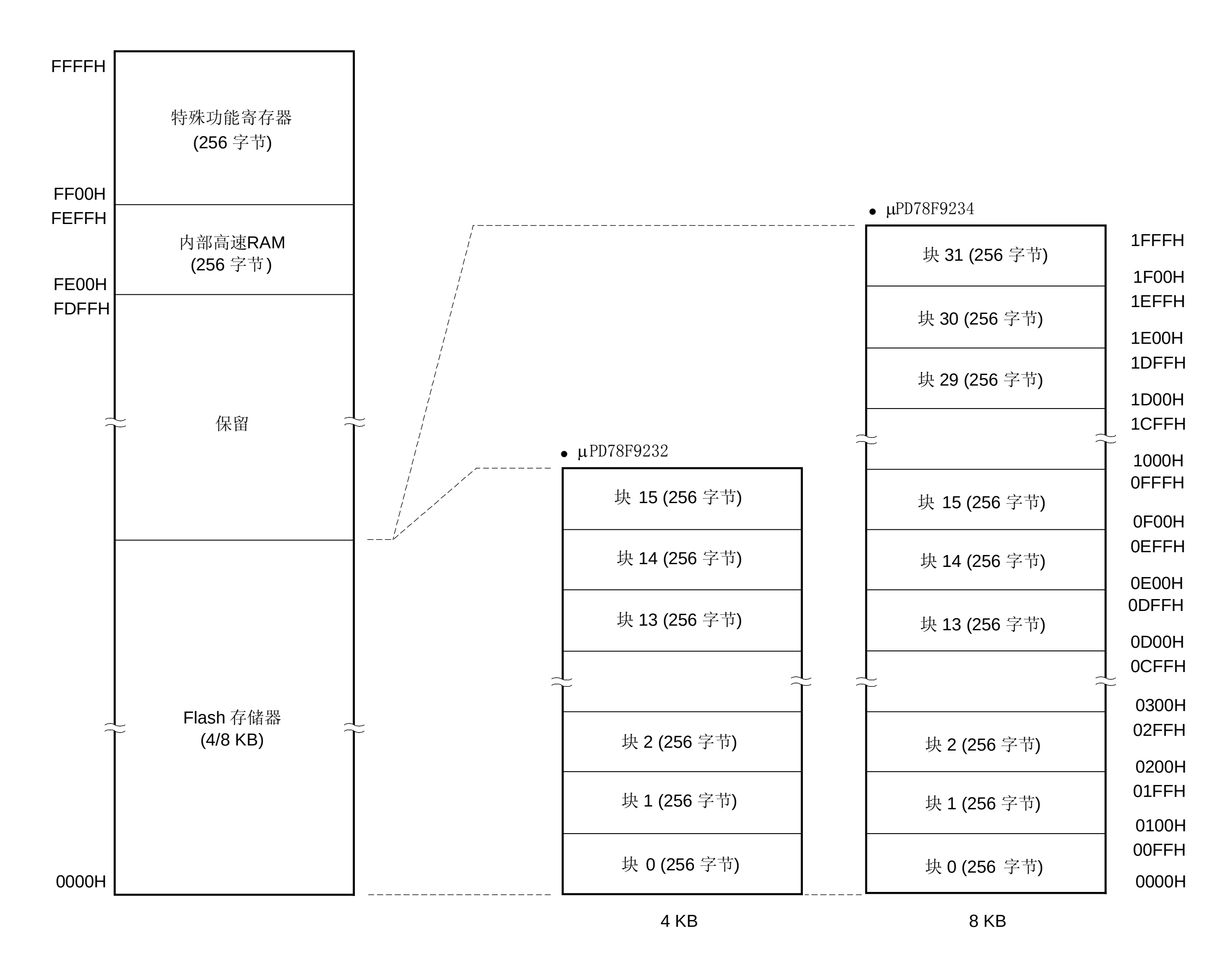
78K0S/KB1+的内部 flash 存储器有如下特点：

- 不需要为擦除/写入准备单独的专用电源
- 容量：4 KB/8 KB
  - 擦除单元：1 块（256 字节）
  - 写入单元：1 字节
- 写入方法
  - 使用专用的 flash 编程器写入（on-board/off-board 编程）
  - 使用用户程序写入(自编程)
- Flash 存储器支持写禁止功能（安全功能）

19.2 存储器配置

4/8 KB 的 flash 内部存储器区域被分为 16/32 个块，并且能够以块为单位写入和擦除数据。所有的块也能够被一次性擦除。

图 19-1. **Flash** 存储器映射图



19.3 功能概要

78K0S/KB1+ 的内部 flash 存储器能够通过专用的 flash 编程器写入，不论 78K0S/KB1+ 是否已经连接到目标板（on-board/off-board 编程）。

提供使用用户程序写入程序的功能(自编程)，当目标系统已经生产/出货后要改变程序时，该功能为用户提供了一种较为理想的解决办法。

另外，支持安全功能，禁止未经授权的人写入内部 flash 存储器中的用户程序。

参考 **19.7.4 安全设置** 部分，有详细的安全功能说明。

表 19-1. 写入方法

写入方法	功能概要	操作模式
On-board 编程	在器件已经安装在目标系统之后，可通过专用 flash 编程器，写入 flash 存储器。	Flash 存储器编程模式
Off-board 编程	在器件安装在目标系统之前，可通过专用 flash 编程器和专用编程适配板，写入 flash 存储器。	
自编程	可通过执行一段预先通过 on-board/off-board 编程方法写入到 flash 存储器的用户程序来写入 flash 存储器。	自编程模式

备注 1. FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

2. flash 存储器写入控制功能的详细论述请参考以下部分。

- 19.7 **On-Board 和 Off-Board Flash 存储器编程。**
- 19.8 **通过自写入实现存储器编程。**

19.4 使用 flash 编程器写入

下列两种专用 flash 编程器能够被用来向 78K0S/KB1+ 的内部 flash 存储器写入数据。

- FlashPro4 (PG-FP4, FL-PR4)
- PG-FPL2

数据能够通过专用 flash 编程器以 on-board 或 off-board 方式写入 flash 存储器中。

(1) On-board 编程

78K0S/KB1+ 在安装到目标系统之后，flash 存储器的内容能够被写入。连接专用 flash 编程器和测试点的连接器必须安装到目标系统上。只有在使用晶体/陶瓷振荡器写入数据时才要求测试点（参考图 19-5 测试点的安装）。

(2) Off-board 编程

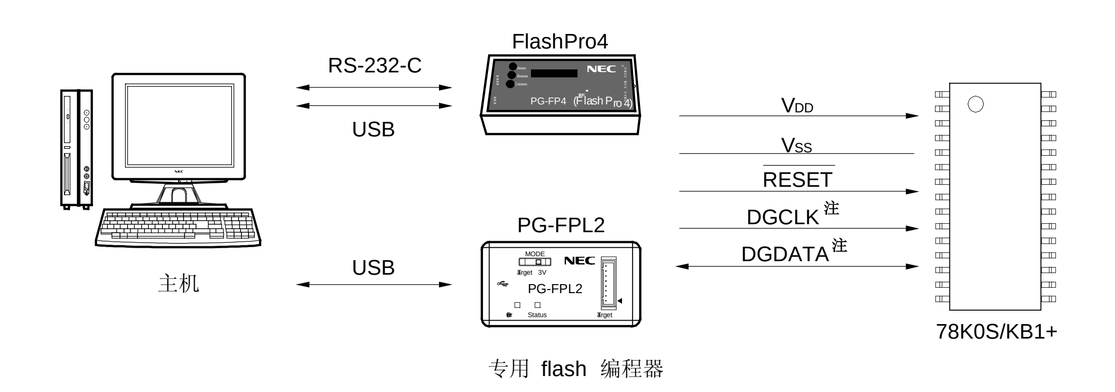
在 78K0S/KB1+ 被安装到目标系统以前，数据可以通过专用编程适配器（FA 系列）写入 flash 存储器中。

备注 FL-PR4 和 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

19.5　编程环境

编程环境要求如下图所示。

图 19-2.　flash 存储器的编程环境



注　　DGCLK 和 DGDATA 是单线双工通讯接口，它们使用 UART 通讯模式。

需要用一台主机来控制专用 flash 编程器。当使用 PG-FP4 或 FL-PR4 时，程序从主机下载到 flash 编程器之后，数据才能被写入。

UART 被用来在专用 flash 编程器与 78K0S/KB1+ 连接时操作写入和擦除。当使用 off-board 方式写入 flash 存储器时，需要一个专用编程适配器(FA 系列)。

可以从网络上下载最新的编程器固件，GUI，和参数文件等开发工具，网址：

(http://www.necel.com/micro/ods/jpn/index.html).

表 19-2.　78K0S/KB1+　和 FlashPro4 之间的连线

FlashPro4 连接引脚			78K0S/KB1+ 连接引脚	
引脚名称	I/O	引脚功能	引脚名称	引脚编号
CLK 注	输出	78K0S/KB1+ 的时钟	X1/P121	8
FLMD0 注	输出	On-board 模式信号		
SI/RxD 注	输入	接收信号	X2/P122	9
SO/TxD 注	输出	接收信号/on-board 模式信号		
/RESET	输出	复位信号	$\overline{\text{RESET}}$/P34	10
V_{DD}	-	V_{DD} 电压产生/电压监视	V_{DD}	7
GND	-	地	V_{SS}	6

注　　在 78K0S/KB1+ 中， CLK 和 FLMD0 信号是连到 X1 引脚的，SI/RxD 和 SO/TxD 信号是连到 X2 引脚的;因此，这些信号需要直接相连起来。

图 19-3. 与 FlashPro4 的连线图

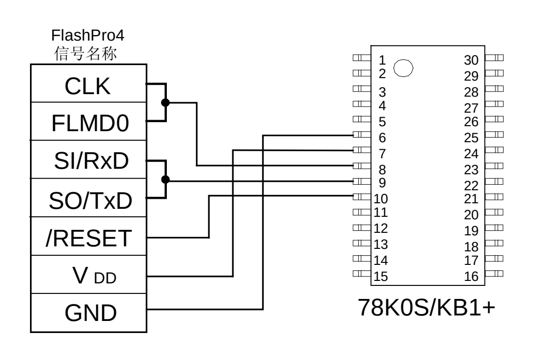
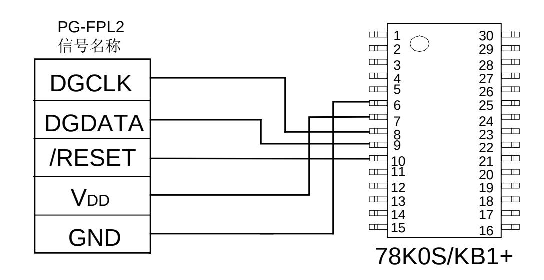


表 19-3. 78K0S/KB1+ 和 PG-FPL2 之间的连线

PG-FPL2 连接引脚			78K0S/KB1+连接引脚	
引脚名称	I/O	引脚功能	引脚名称	引脚编号
CLK	输出	78K0S/KB1+ 的时钟	X1/P121	8
DGDATA	I/O	发送/接收信号，on-board 模式信号	X2/P122	9
/RESET	输出	复位信号	$\overline{\text{RESET}}$/P34	10
V_{DD}	I/O	V_{DD} 电压产生	V_{DD}	7
GND	-	地	V_{SS}	6

图 19-4. 与 PG-FPL2 的连线图



19.6 On-Board 引脚连接

如要以 on-board 方式写入 flash 存储器，目标系统必须提供连接到专用 flash 编程器的连接器。首先目标板要可以选择正常操作模式或者 flash 存储器写入模式。

当设定为 flash 存储器写入模式时，所有不参与写入 flash 存储器的引脚与复位后的状态相同。因此。如果外部器件不能在复位后识别这个状态，引脚必须以如下方式进行处理。

在自写入模式下引脚的状态和 HALT 模式的状态是相同的。

19.6.1 X1 和 X2 引脚

X1 和 X2 引脚作为 flash 存储器编程的串行接口。因此，如果当 X1 和 X2 引脚连接外部设备时，将会产生信号冲突。为了阻止信号发生冲突，应使引脚 X1 和 X2 与外部设备分开连接。

当使用晶体或陶瓷振荡器作为系统时钟且隔离振荡器比较困难时，如要在已安装振荡器的情况下使用 on-board 写入方式须采用下列方法(1) 和 (2)。

(1) 在器件与振荡器之间加入最少的测试点，通过测试点连接 flash 编程器。两者连线尽可能要短（参考图 19-5 和表 19-4）。

(2) 可以使用专用 flash 编程器的 GUI 软件来设定数据写入时通讯时钟的振荡器频率。查找串/并行振荡器和振荡器使用的反振频率，设定振荡器频率，以便使振荡频率在±10%的范围之外（参考图 19-6 和表 19-5）。

图 19-5. 安装测试点的示例

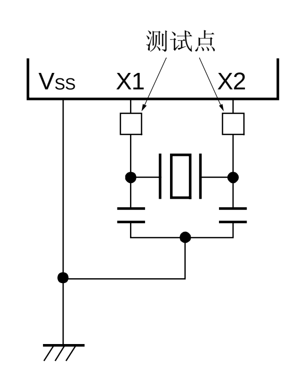


表 19-4. 使用的时钟和测试点的安装

使用的时钟		安装测试点
高速 Ring-OSC 时钟		不需要
外部时钟		
晶体/陶瓷振荡器时钟	在振荡器安装之前	
	在振荡器安装之后	需要

图 19-6. PG-FP4 GUI 软件设置示例

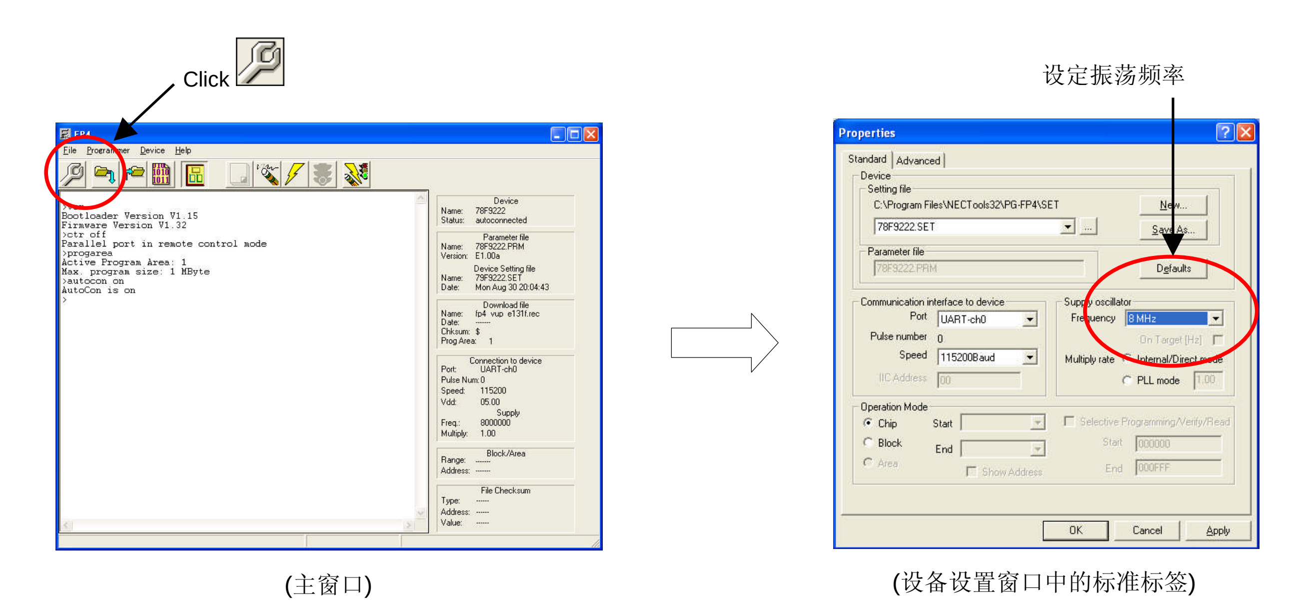


表 19-5. 振荡频率和 PG-FP4 GUI 软件设置值示例

振荡频率	PG-FP4 GUI 软件设置值示例（通信频率）
1 MHz ≤ fx < 4 MHz	8 MHz
4 MHz ≤ fx < 8 MHz	9 MHz
8 MHz ≤ fx < 9 MHz	10 MHz
9 MHz ≤ fx ≤ 10 MHz	8 MHz

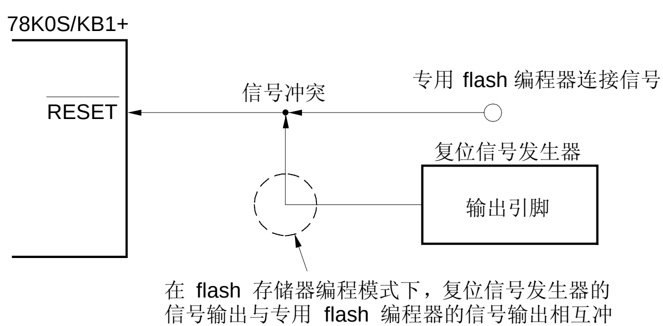
注意事项　　以上值为推荐值。由于环境不同，这些值可能改变，所以使用时请先估计一个值。

19.6.2 $\overline{\text{RESET}}$ 引脚

若专用 flash 编程器的复位信号与连接到目标板上的信号发生器的 $\overline{\text{RESET}}$ 引脚相连，将会发生信号冲突。为了防止这种冲突发生，需要隔离复位信号发生器。

如果在 flash 存储器编程模式下从用户系统输入复位信号，则不能够对 flash 存储器正确编程。因此不要输入除专用 flash 编程器复位信号外的任何信号。

图 19-7.　信号冲突（$\overline{\text{RESET}}$ 引脚）



19.6.3　端口引脚

当 flash 存储器编程模式设定后，所有不参与写入的引脚都进入与复位后相同的状态。如果连接这些端口引脚的外部设备不能在复位后立即识别这个状态，则此引脚必须通过一个电阻连接到 V_{DD} 或 V_{SS} 上。

引脚的状态在自编程模式下和 HALT 模式下是相同的。

19.6.4　电源供应

将 V_{DD} 引脚连接到 flash 编程器的 V_{DD} 引脚，将 V_{SS} 引脚连接到 flash 编程器的 V_{SS} 引脚。

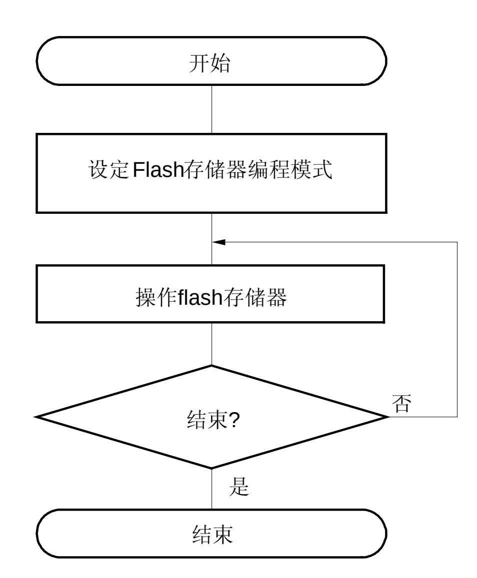
采用与正常工作模式下一样的电源供应（AV_{REF} 和 AV_{SS} ）。

19.7　On-Board 和 Off-Board Flash 存储器编程

19.7.1　对 flash 存储器的控制

操作 flash 存储器的步骤如下图所示。

图 19-8.　Flash 存储器的操作步骤



19.7.2 Flash 存储器编程模式

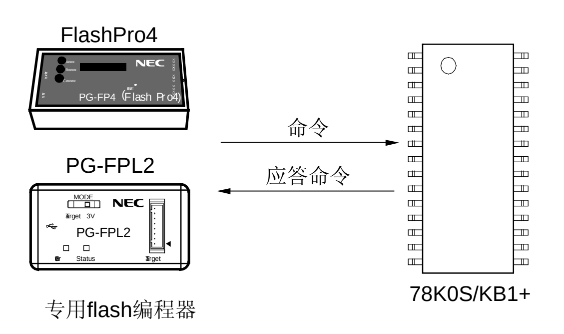
为了使用专用 flash 编程器写入 flash 存储器里的内容，需要将 78K0S/KB1+ 设定在 flash 存储器编程模式。当 78K0S/KB1+ 连接到 flash 编程器并且通讯命令发送到微控制器时，微控制器设定在 flash 存储器编程模式。

当以 on-board 方式写入 flash 存储器时，可使用跳线改变操作模式。

19.7.3 通讯命令

通过命令使 78K0S/KB1+与专用 flash 编程器通讯。从 flash 编程器发送到 78K0S/KB1+的信号称为命令，从 78K0S/KB1+ 发送到 flash 编程器的命令称为应答命令。

图 19-9. 通讯命令



下表列出了 78K0S/KB1+ 的 flash 存储器通讯命令。这些命令都是由编程器发出，78K0S/KB1+ 根据相应命令完成处理。

表 19-6. Flash 存储器控制命令

分类	命令名称	功能
擦除	一次性擦除(芯片擦除)命令	擦除整个存储器的内容
	块擦除命令	擦除指定块的内容
写入	写入命令	将数据写入指定地址范围中，并进行校验
校验和	校验和命令	读取指定地址范围的校验和，并与写入数据进行比较
块检测	块检测命令	确定整个存储器的擦除状态
安全	安全设置命令	禁止一次性擦除(芯片擦除)命令，块擦除命令，和写入命令以阻止第三方操作

78K0S/KB1+ 对专用 flash 编程器发出的命令将会返回一个应答命令，应答命令如下表所示：

表 19-7. 应答命令

命令名称	功能
ACK	对命令/数据的应答
NAK	对无效命令/数据的应答

19.7.4 安全设置

通过使用安全设置命令能够禁止以下操作。

注意事项 当编程模式下一次被设置时，安全设置为有效。因此，当执行安全设置命令时，先退出编程模式，然后再次设置编程模式。

- 一次性擦除(芯片擦除)
  禁止对 flash 存储器中的整个块执行块擦除和一次性擦除(芯片擦除)命令。一旦一次性擦除(芯片擦除)命令被禁止，所有的保护设定将不能取消。

  注意事项 在一次性擦除(芯片擦除)的安全设置设定之后，擦除指令不能执行。另外，即使执行写入命令，与已写入 flash 存储器不同的数据是不能被写入的，因为擦除命令已被禁止。

- 块擦除
  禁止对 flash 存储器中的某个指定块执行块擦除命令。该命令设置可通过一次性擦除(芯片擦除)命令取消。

- 写
  禁止对 flash 存储器中的整块执行写入和块擦除命令。该命令设置可通过一次性擦除(芯片擦除)命令取消。

flash 存储器在默认设置下，一次性擦除(芯片擦除)，块擦除和写入命令是有效的。以上安全设定只是在 on-board/off-board 编程时有效。每一个安全设定都能够和其他设定一起使用。

表 19-8 显示了当允许 78K0S/KB1+ 的安全功能后，擦除与写入命令的关系。

表 19-8. 当设置安全功能后命令之间的关系

安全 \ 命令	一次性擦除(芯片擦除)	块擦除命令	写命令
当一次性擦除(芯片擦除)安全操作有效时	禁止	禁止	有效[注]
当块擦除安全操作有效时	有效		有效
当写入命令安全操作有效时			禁止

注 因为擦除命令被禁止，所以与已写入 flash 存储器不同的数据是不能被写入的

表 19-9 显示在每个编程模式下安全设置与操作的关系。

表 19-9. 在每个编程模式下安全设置与操作的关系

编程模式 / 安全设置	On-Board/Off-Board 编程		自编程	
	安全设置	安全操作	安全设置	安全操作
一次性擦除(芯片擦除)	允许	有效注1	不允许	无效注2
块擦除				
写入				

注 1. 安全设定可以禁止任何命令的执行。
2. 不管安全设定如何，自编程命令都可以执行。

19.8 通过自写入实现存储器编程

78K0S/KB1+ 支持自编程功能，可以通过用户程序来写入 flash 存储器，因此具有在线升级程序的功能。

注意事项 在执行自写入以前，自写入处理程序必须先被包含在用户程序中。

备注 若想用 78K0S/KB1+ 的内部 flash 存储器当作外部 EEPROM 来存储数据时,请参考"78K0S/Kx1+ EEPROM Emulation AN" (U17379E)。

19.8.1 自编程概述

如要执行自编程， 需将工作模式由用户程序的正常操作(正常模式)切换到自编程模式。在自编程模式下，先设置特定寄存器，通过执行 HALT 指令执行 flash 存储器的擦写处理程序。当程序结束时，将自动释放 HALT 状态。

如要切换到自编程模式，需以特定顺序对特定寄存器进行操作，详情请参考 19.8.4 **从普通模式切换到自编程模式示例**。

备注 自编程写入数据可参照 MOV 指令。

表 19-10. 自编程模式

模式	用户程序执行	通过 HALT 指令对 flash 存储器执行写入/擦除命令
普通模式	允许	–
自编程模式	允许注	允许

注 在自编程模式下，可屏蔽中断处理服务程序被禁止。

图 19-10 是自编程框图，图 19-11 是自编程状态转换图，表 19-11 列举了自编程控制命令。

图 19-10. 自编程框图

内部总线

Flash 保护命令寄存器 (PFCMD)

Flash 编程模式控制寄存器 (FLPMC)

自编程模式设定寄存器

自编程模式设定次序

HALT 信号

执行自编程命令

HALT 释放信号

Flash 存储控制器

验证电路

写入电路

擦除电路

Flash 存储器

Flash 状态寄存器 (PFS)

WEPRERR | VCERR | FPRERR

不相等

Flash 写缓冲寄存器 (FLW)

flash 编程命令寄存器 (FLCMD)

FLCMD2 | FLCMD1 | FLCMD0

3

保护字节

PRSELF4 | PRSELF3 | PRSELF2 | PRSELF1 | PRSELF0

5

累加电路

Flash 地址指针 L (FLAPL)

相等

Flash 地址指针 L 比较寄存器 (FLAPLC)

Flash 地址指针 H (FLAPH)

相等

Flash 地址指针 H 比较寄存器 (FLAPHC)

内部总线

图 19-11.　自编程状态转换图

表 19-11.　自编程控制命令

命令名称	功能	从 HALT 指令执行开始到命令执行结束的时间
内部校验	该指令用于验证数据是否已经被正确的写入 flash 存储器中，数据写入后，指定块的序号，开始和结束地址，然后执行这个指令。	对于 1 个块的内部校验：6.8ms(内部校验命令执行一次) 对于 1 个字节的内部校验：27μs
块擦除	该指令用于擦除指定块的内容，先指定块的序号，然后执行此命令。	8.5 ms
空白块检验	该指令用于校验指定块是否已被擦除。先指定块的序号，然后执行此命令。	480 μs
字节写	该指令用于对 flash 存储器指定地址写入 1 字节数据。在确定写入地址和写入数据之后，就可执行此命令。	150 μs

19.8.2　自编程功能的注意事项

- 如果在自编程期间产生中断，则中断请求标志被置为“1”，在自编程模式释放后才执行中断服务程序。为了避免这种情况，在自编程期间或以特定顺序从普通模式切换到自编程模式之前禁止中断服务（通过设置 MK0 和 MK1 为 FFH 并执行 DI 指令）。
- 当自编程命令执行时，不能执行其它命令。因此，要先清除和重启看门狗定时器计数器，以便在自编程时计数器不会溢出。关于执行自编程需要的时间请参考表 19-11。
- 当执行自编程命令时，不使用 RAM。
- 当 flash 存储器正在被写入或擦除时，如果供电电压下降或者有复位信号输入，则写入或擦除操作不能得到保证。
- 在块擦除期间空白数据的设置值设为 0FFH。
- 当选用振荡器或外部时钟作为主时钟时，从自编程模式的设定到 HALT 指令的执行需要 16μs 的等待时间。
- 自编程模式下和 HALT 模式下引脚的状态是相同的。
- 因为在自编程模式中，禁止通过 on-board/off-board 编程设置安全功能，所以无论安全功能如何设定，自编程命令都可执行。想要在自编程期间禁止写入或擦除进程，则需要设定保护字节。
- 当执行自编程命令时，如果 flash 地址指针 H(FLAPH)和 flash 地址指针 H 比较寄存器(FLAPHC)的第 4 位到第 7 位为 1，在执行自编程命令之前一定要对这些位清零。

19.8.3　自编程功能所使用的寄存器

自编程功能使用以下的寄存器：

- Flash 编程模式控制寄存器 (FLPMC)
- Flash 保护命令寄存器 (PFCMD)
- Flash 状态寄存器 (PFS)
- Flash 编程命令寄存器 (FLCMD)
- Flash 地址指针 H 和 L (FLAPH 和 FLAPL)
- Flash 地址指针 H 比较寄存器和 flash 地址指针 L 比较寄存器(FLAPHC 和 FLAPLC)
- Flash 写缓冲寄存器 (FLW)

78K0S/KB1+ 在 flash 存储器的 0081H 地址有一个区域被称为保护字节。

(1) Flash 编程模式控制寄存器(FLPMC)

这个寄存器用于在自编程模式下向 flash 存储器写入数据时设定工作模式，并可用于读取保护字节的设定值。

只有按特定顺序操作(参考 **19.8.3 (2) flash 保护命令寄存器(PFCMD)**) 数据才能写入 FLPMC，如此就不会因为偶然的噪声干扰或者程序挂起等故障导致应用系统意外停止。

该寄存器可通过一个 8 位的存储操作指令来设置。

复位后该寄存器内容不确定。

图 19-12.　Flash 编程模式控制寄存器(FLPMC)的格式

地址：　FFA2H　　复位后：　不确定[注 1]　　R/W[注 2]

符号	7	6	5	4	3	2	1	0
FLPMC	0	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	0	FLSPM

FLSPM	自编程模式期间操作模式的选择
0	普通模式 flash存储器指令可从任一地址取出。
1	自编程模式 在执行HALT指令之前，首先要设定命令、地址偏移量、写入数据并设置FLSPM为1。在完成以上操作后，执行HALT指令，然后flash存储器的模式将从普通模式转为编程模式。

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	保护字节的设定值可通过这些位读出。

注　1.　当复位释放时，第 0 位(FLSPM)被清 0。当复位释放后，保护字节的设定值可从第 2 位到第 6 位（PRSELF0 到 RESELF4）读出。

2.　第 2 位到第 6 位（PRSELF0 到 RESELF4）是只读的。

注意事项 1.　在设置自编程模式时请注意以下几点。

- 如果在自编程期间产生中断，则中断请求标志被置为“1”，在自编程模式释放后才执行中断服务程序。为了避免这种情况，在自编程期间或以特定顺序从普通模式切换到自编程模式之前禁止中断服务（通过设置 MK0 为 FFH 并执行 DI 指令）。
- 当自编程命令执行时，不能执行指令。因此，要先清除和重启看门狗定时器计数器，以便在自编程时计数器不会溢出。关于执行自编程需要的时间请参考表 19-11。
- 当 flash 存储器正在写入或擦除时，如果供电电压下降或者有复位信号输入，则写入或擦除操作不能得到保证。

2.　当选择振荡器或外部时钟作为主时钟时，从 FLSPM 被设置为 1 到 HALT 指令的执行需要 16 μs 的等待时间。

(2) Flash 保护命令寄存器 (PFCMD)

如果由于噪声或程序挂起等原因引起的故障导致应用系统停止，此时对 flash 编程模式控制寄存器(FLPMC)进行写入操作将对系统产生严重影响。PFCMD 用来保护 FLPMC 的写入，这样就不会使应用系统由于疏忽导致停止。

必须按以下顺序对 FLPMC 执行写入操作。

〈1〉 写入一个特定值到 PFCMD (PFCMD = A5H)

〈2〉 写这个值给 FLPMC (此时写入操作是无效的)

〈3〉 写这个值的反码给 FLPMC (此时写入操作是无效的)

〈4〉 写这个值给 FLPMC (此时写入操作是有效的)

这样重复写入寄存器，可使寄存器不能被非法写入。

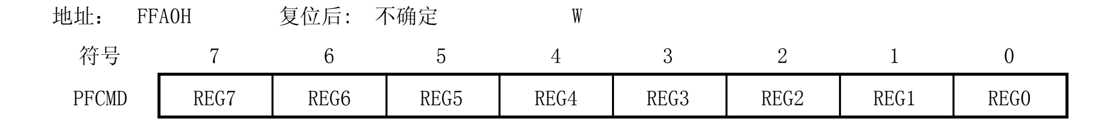
可以通过 flash 状态寄存器(PFS)的第 0 位(FPRERR)来验证是否出现非法的存储操作。

每当 FLPMC 的值发生改变的时候都必须向 PFCMD 写入 A5H。

PFCMD 的值可通过一个 8 位的存储器操作指令来设置。

复位后寄存器 PFCMD 不确定。

图 19-13. Flash 保护命令寄存器 (PFCMD)的格式



注意事项 **当以特定顺序执行操作时禁止中断服务（通过设置 MK0 和 MK1 为 FFH 并执行 DI 指令）。**

(3) Flash 状态寄存器 (PFS)

如果数据不能按正确顺序(写入 flash 保护命令寄存器 PFCMD)写入受保护的 flash 编程模式控制寄存器(FLPMC)中，FLPMC 不能被写入并且出现保护错误，PFS 的第 0 位(FPRERR)将被置 1。

当 FPRERR 为 1 时，可以通过直接写 0 来清 0。

自编程模式期间可能产生的错误能够通过 PFS 的第 1 位(VCERR)和第 2 位(WEPRERR)位反应出来，(VCERR)和(WEPRERR)可以通过写 0 来清 0。

要验证操作是否执行正确，PFS 寄存器的所有标志位必须被预先清零。

PFS 可以使用 1 位或 8 位存储器操作指令设置。

复位后 PFS 被置为 00H。

图 19-14. Flash 状态寄存器(PFS)的格式

地址: FFA1H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	WEPRERR	VCERR	FPRERR

1. FPRERR 标志的操作条件

〈设置条件〉

- 刚执行的对外围寄存器的写入命令没有给PFCMD写入特定值（PFCMD = A5H）时，对PFCMD进行写入操作。
- 在〈1〉之后，如果第一条存储指令操作的是外围寄存器而不是 FLPMC。
- 在〈2〉之后，如果第一条存储指令操作的是外围寄存器而不是 FLPMC。
- 在〈2〉之后，如果第一条存储指令所写的值不同于 FLPMC 值的反码。
- 在〈3〉之后，如果第一条存储指令操作的是外围寄存器而不是 FLPMC。
- 在〈3〉之后，如果第一条存储指令所写的值不同于 FLPMC 值(在〈2〉中所写)。

备注 以上角括号中的数字对应于(2)Flash 保护命令寄存器(PFCMD)。

〈复位条件〉

- 如果 FPRERR 标志被写入 0
- 如果复位信号输入

2. VCERR 标志的操作条件

〈设置条件〉

- 擦除校验错误
- 内部写入校验错误

如果 VCERR 被设为 1，则说明 flash 没有被正确的擦除或写入。在指定的程序里要再次对 flash 进行擦除或写入。

备注 当发生擦除或写入保护错误时，VCERR 标志可能也会被置 1。

〈复位条件〉

- VCERR 标志被置为 0
- 外部复位信号输入

3. WEPRERR 标志操作条件

〈设置条件〉

- 使用保护字节指定一个区域防止该区域因 flash 地址指针 H (FLAPH)和执行相关命令而引起的数据擦除或写入。

〈复位条件〉

- WEPRERR 标志被置 0
- 外部复位信号输入

(4) Flash 编程命令寄存器(FLCMD)

在自编程模式下，FLCMD 被用来确定 flash 的擦除、写入、校验操作。

该寄存器可通过 1 位或 8 位的操作指令来设置。

复位后寄存器值置为 00H。

图 19-15. Flash 编程命令寄存器 (FLCMD)的格式

地址: FFA3H　　复位后: 00H　　R/W

符号	7	6	5	4	3	2	1	0
FLCMD	0	0	0	0	0	FLCMD2	FLCMD1	FLCMD0

FLCMD2	FLCMD1	FLCMD0	命令名称	功能
0	0	1	内部校验	此命令用来验证写入flash内的数据是否正确。数据写入后，指定块的序号、开始和结束地址，运行这个命令。如果错误发生，flash状态寄存器(PFS)的第1位(VCERR)或第2位(WEPRERR)将被置为1。
0	1	1	块擦除	此命令用于对指定块的擦除。它可以用在on-board和自编程模式下。
1	0	0	空白块检测	此命令用于检测指定块是否已经被擦除。
1	0	1	字节写	此命令用于将一个字节数据写入flash存储器的指定地址。指令执行前要指定写入地址和写入数据。
其他[注]			禁止设置	

注 如果不是以上列举的值并且工作模式设置为自编程模式，则自编程模式被立刻取消且不执行任何操作。此时，不会设置 PFS 寄存器的标志。

(5) Flash 地址指针 H 和 L (FLAPH 和 FLAPL)

Flash 地址指针 H 和 L (FLAPH 和 FLAPL)用于在自编程模式下指定 flash 的删除、写入、验证的开始地址。
FLAPH 和 FLAPL 由计数器组成，当编程命令不执行时，它们递增直到与 FLAPHC 和 FLAPLC 的值相等。因此，当程序命令执行时，PLAPHC 和 FLAPLC 的值需要重新设置。
Flash 地址指针 H 和 L (FLAPH 和 FLAPL)可通过 1 位或 8 位操作指令来设置。
复位后寄存器值不确定。

图 19-16.　Flash 地址指针 H/L (FLAPH/FLAPL)格式

地址: FFA4H，FFA5H　复位后: 不确定　R/W

FLAPH (FFA5H)								FLAPL (FFA4H)							
0	0	0	0	FLAP11	FLAP10	FLAP9	FLAP8	FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0

注意事项　　**执行自编程命令之前，要确保 FLAPH 的第 4 位到第 7 位和 FLAPHC 清零。**

(6) Flash 地址指针 H 比较寄存器和 flash 地址指针 L 比较寄存器 (FLAPHC 和 FLAPLC)

在自编程模式下校验 flash 存储器时，flash 地址指针比较寄存器 H 和 L(FLAPHC 和 FLAPLC)用于指定内部序列操作的地址范围。
将 FLAPH 的值赋给 FLAPHC。将 flash 存储器校验范围的最末端地址赋给 FLAPLC。
FLAPHC 和 FLAPLC 使用 1 位或 8 位存储器操作指令来设置。
复位后 FLW 寄存器值置为 00H。

图 19-17.　Flash 地址指针 H/L 比较寄存器 (FLAPHC/FLAPLC)的格式

地址: FFA6H，FFA7H　复位后: 00H　R/W

FLAPHC (FFA7H)								FLAPLC (FFA6H)							
0	0	0	0	FLAPC11	FLAPC10	FLAPC9	FLAPC8	FLAPC7	FLAPC6	FLAPC5	FLAPC4	FLAPC3	FLAPC2	FLAPC1	FLAPC0

注意事项
1. **在自编程命令执行以前，确保 flash 地址指针 H(FLAPH)的第 4 位至第 7 位和 flash 地址指针 H 比较寄存器(FLAPHC)清零。**
2. **当进行块擦除、校验、空白检测时，将块的序号(与 FLAPH 的值相同)赋值给 FLAPHC。**
3. **当块擦除完成时 FLAPLC 要清零，当块空白检测完成后 FLAPLC 要设为 FFH。**

(7) Flash 写缓冲寄存器(FLW)

该寄存器用于存储要写入 flash 存储器的数据。

该寄存器值可用 1 位或 8 位存储器操作指令设置。

复位后 FLW 寄存器值置为 00H。

图 19-18. flash 写缓冲寄存器(FLW)的格式

地址： FFA8H 复位后： 00H R/W

符号	7	6	5	4	3	2	1	0
FLW	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

(8) 保护字节

保护字节用于指定防止写入或擦除的区域。该指定的保护区域只在自编程模式下有效。因为受保护区域在自编程模式下是不可用的，所以保护区域内的数据是受保护的。

图 19-19. 保护字节的格式(1/2)

地址： 0081H

7	6	5	4	3	2	1	0
1	PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	1	1

- μPD78F9232

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	1	0	0	0	块15到0是受保护的
0	1	0	0	1	块13到0是受保护的 块14到15可被写入或擦除
0	1	0	1	0	块11到0是受保护的 块12到15可被写入或擦除
0	1	0	1	1	块9到0是受保护的 块10到15可被写入或擦除
0	1	1	0	0	块7到0是受保护的 块8到15可被写入或擦除
0	1	1	0	1	块5到0是受保护的 块6到15可被写入或擦除
0	1	1	1	0	块3到0是受保护的 块4到15可被写入或擦除
0	1	1	1	1	块1到0是受保护的 块2到15可被写入或擦除
1	1	1	1	1	所有的块都可被写入或擦除
其它					禁止设置

图 19-19. 保护字节的格式(2/2)

• μPD78F9234

PRSELF4	PRSELF3	PRSELF2	PRSELF1	PRSELF0	状态
0	0	0	0	0	块31到0是受保护的
0	0	0	0	1	块29到0是受保护的 块30到31可被写入或擦除
0	0	0	1	0	块27到0是受保护的 块28到31可被写入或擦除
0	0	0	1	1	块25到0是受保护的 块26到31可被写入或擦除
0	0	1	0	0	块23到0是受保护的 块24到31可被写入或擦除
0	0	1	0	1	块21到0是受保护的 块22到31可被写入或擦除
0	0	1	1	0	块19到0是受保护的 块20到31可被写入或擦除
0	0	1	1	1	块17到0是受保护的 块18到31可被写入或擦除
0	1	0	0	0	块15到0是受保护的 块16到31可被写入或擦除
0	1	0	0	1	块13到0是受保护的 块14到31可被写入或擦除
0	1	0	1	0	块11到0是受保护的 块12到31可被写入或擦除
0	1	0	1	1	块9到0是受保护的 块10到31可被写入或擦除
0	1	1	0	0	块7到0是受保护的 块8到31可被写入或擦除
0	1	1	0	1	块5到0是受保护的 块6到31可被写入或擦除
0	1	1	1	0	块3到0是受保护的 块4到31可被写入或擦除
0	1	1	1	1	块1到0是受保护的 块2到31可被写入或擦除
1	1	1	1	1	所有的块都可被写入或擦除
其它					禁止设置

19.8.4　从普通模式切换到自编程模式的示例

在执行自编程模式前必须由普通模式切换至自编程模式。

切换至自编程模式的操作过程示例如下：

〈1〉　如果使用了中断功能则必须首先屏蔽中断（通过设置的中断屏蔽标志寄存器(MK0, MK1)为 FFH 并执行 DI 指令)）。

〈2〉　清除 flash 状态寄存器（PFS）。

〈3〉　按指定顺序设置自编程模式。

- 设定 PFCMD 值为 A5H。
- 设定 FLPMC 值为 01H (此时写入操作是无效的)。
- 设定 FLPMC 值为 0FEH (01H 的反码)(此时写入操作是无效的)。
- 设定 FLPMC 值为 01H (此时写入操作是有效的)。

〈4〉　用 PFS 的第 0 位(FPRERR)检验指定顺序执行的结果是否正确。

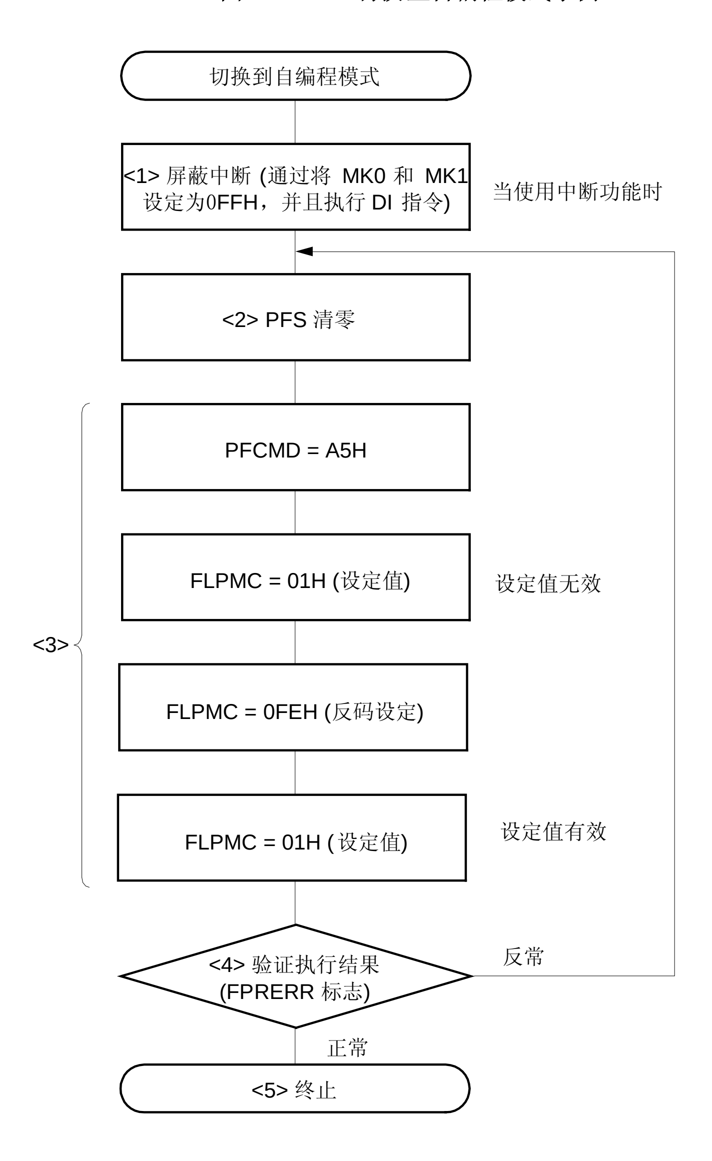
错误 → 〈2〉

正常 → 〈5〉

〈5〉　模式转换结束。

注意事项　在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除和写入操作发生。

图 19-20.　切换至自编程模式示例



注意事项　在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除和写入操作发生。

备注　图 19-20 中的〈1〉到〈5〉 对应于 19.8.4 中的〈1〉到〈5〉 （前页)。

以下是切换到自编程模式的示例程序。

```
;----------------------------
;START
;----------------------------
        MOV        MK0,#11111111B        ; Masks all interrupts
        MOV        MK1,#11111111B

        DI

ModeOnLoop:
        MOV        PFS,#00H
        MOV        PFCMD,#0A5H           ; PFCMD register control
        MOV        FLPMC,#01H            ; FLPMC register control (sets value)
        MOV        FLPMC,#0FEH           ; FLPMC register control (inverts set value)
        MOV        FLPMC,#01H            ; Sets self programming mode with FLPMC register
                                         ; control (sets value)

        MOV        A,PFS
        CMP        A,#00H
        BNZ        $ModeOnLoop           ; Checks completion of write to specific registers
                                         ; Repeats the same processing when an error occurs.

;----------------------------
;END
;----------------------------
```

19.8.5　从自编程模式切换到普通模式的示例

自编程完成后，操作模式一定要从自编程返回到正常模式。

以下是切换到普通模式的过程。

〈1〉　清除状态寄存器（PFS）

〈2〉　按指定的顺序设置普通模式

- 设定 PFCMD 值为 A5H
- 设定 FLPMC 值为 00H (此时写入操作是无效的)
- 设定 FLPMC 值为 0FFH (00H 的反码)(此时写入操作是无效的)
- 设定 FLPMC 值为 00H　(此时写入操作是有效的)

〈3〉　用 PFS 的第 0 位(FPRERR)检验指定顺序执行的结果是否正确

　　错误 → 〈1〉

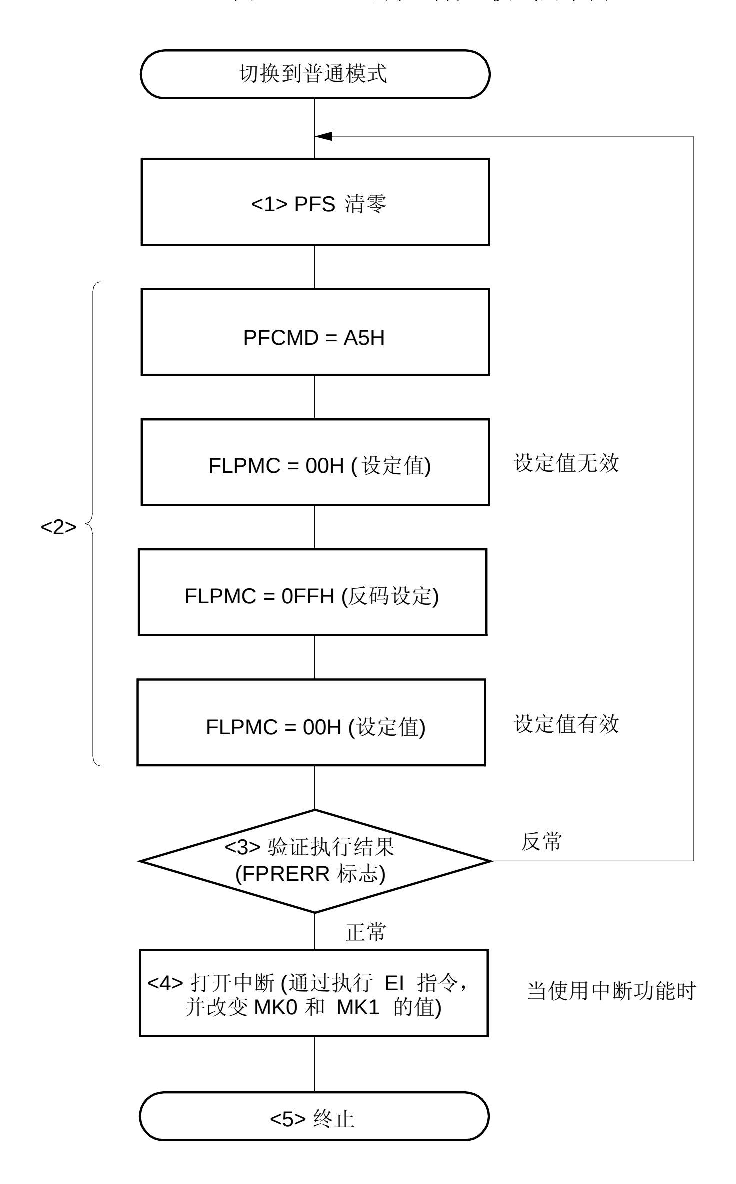
　　正常 → 〈4〉

〈4〉　开放中断处理服务（执行 EI 指令和改变 MK0 的值）以恢复到原来的状态

〈5〉　模式转换结束

注意事项　在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除和写入操作发生。

图 19-21. 切换到普通模式的示例



注意事项 在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除和写入操作发生。

备注 图 19-21 中的〈1〉到〈5〉 对应于 19.8.5 中的〈1〉到〈5〉 （前页）。

以下是切换到普通模式的示例程序。

```
;----------------------------
;START
;----------------------------

ModeOffLoop:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H         ; PFCMD register control
        MOV     FLPMC,#00H          ; FLPMC register control (sets value)
        MOV     FLPMC,#0FFH         ; FLPMC register control (inverts set value)
        MOV     FLPMC,#00H          ; Sets normal mode via FLPMC register control (sets value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOffLoop        ; Checks completion of write to specific registers
                                    ; Repeats the same processing when an error occurs

        MOV     MK0,#INT_MK0        ; Restores interrupt mask flag
        MOV     MK1,#INT_MK1

        EI

;----------------------------
;END
;----------------------------
```

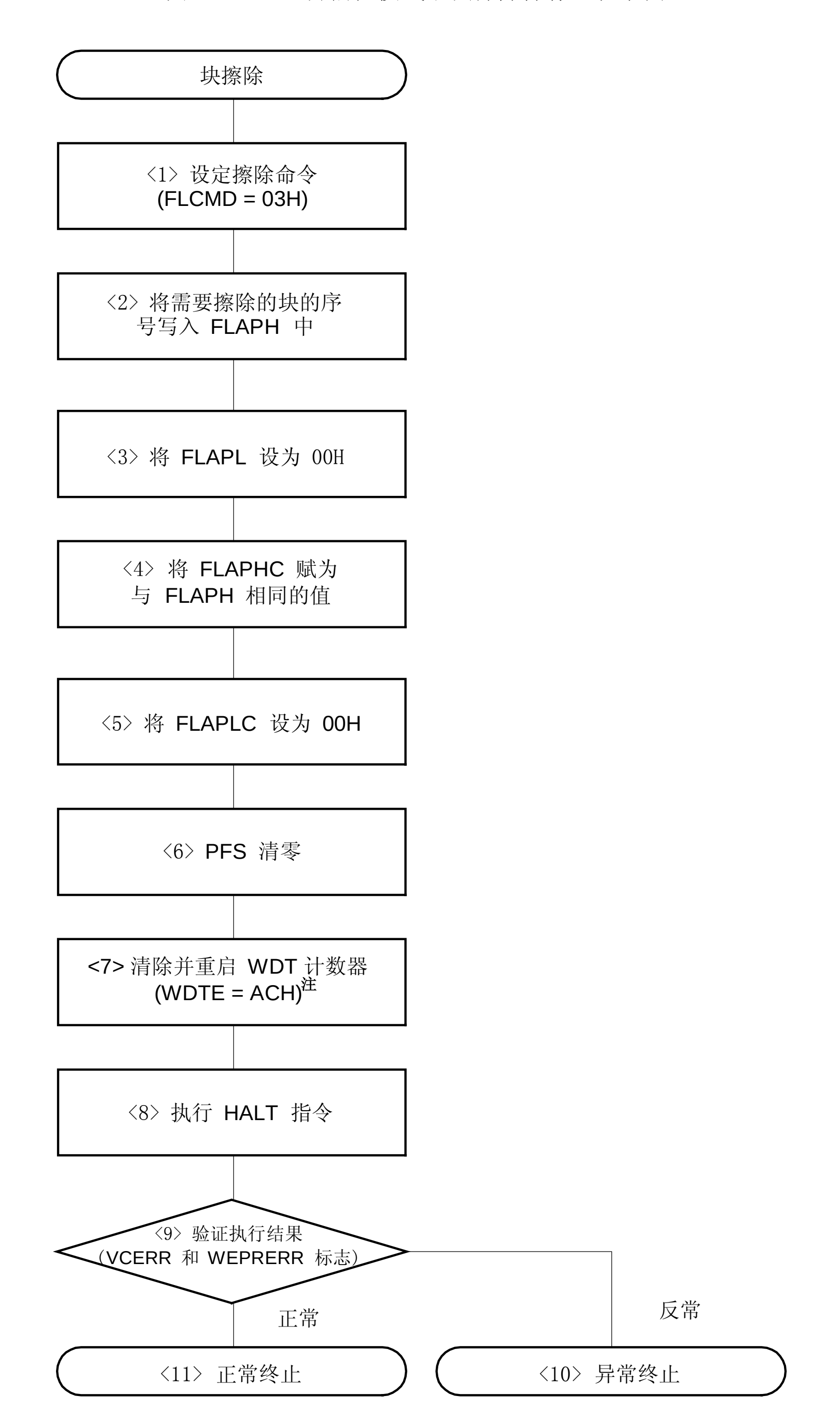
19.8.6　自编程模式下块擦除操作示例

自编程模式下块擦除操作过程如下：

〈1〉 设置编程命令寄存器(FLCMD)值为 03H（块擦除）。
〈2〉 将需要擦除的块的序号值放到 flash 地址指针 H(FLAPH)中。
〈3〉 设置地址指针 L(FLAPL)为 00H。
〈4〉 将 FLAPH 的值写到地址指针 H 比较寄存器(FLAPHC)中。
〈5〉 设置地址指针 L 比较寄存器(FLAPLC)值为 00H。
〈6〉 清除状态寄存器（PFS）。
〈7〉 写 ACH 到看门狗定时器开放寄存器(WDTE)（清零并重启看门狗定时计数器）[注]。
〈8〉 执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
〈9〉 用 PFS 的第 1 位(VCERR)和第 2 位(WEPRERR)检查自编程是否产生错误。
错误 → 〈10〉
正常 → 〈11〉
〈10〉 块擦除处理非正常结束。
〈11〉 块擦除处理正常的结束。

注　当看门狗定时器不使用时无需对其进行设置。

图 19-22. 自编程模式下块擦除操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 19-22 中的〈1〉到〈11〉 对应于 19.8.6 中的〈1〉到〈11〉 （前页）。

以下是自编程模式下块擦除程序示例。

```
;----------------------------
;START
;----------------------------

FlashBlockErase:
        MOV     FLCMD,#03H      ; Sets flash control command (block erase)
        MOV     FLAPH,#07H      ; Sets number of block to be erased (block 7 is specified here)
        MOV     FLAPL,#00H      ; Fixes FLAPL to “00H”
        MOV     FLAPHC,#07H     ; Sets erase block compare number (same value as that of FLAPH)
        MOV     FLAPLC,#00H     ; Fixes FLAPLC to “00H”

        MOV     PFS,#00H        ; Clears flash status register
        MOV     WDTE,#0ACH      ; Clears & restarts WDT
        HALT                    ; Self programming is started
        MOV     A,PFS
        MOV     CmdStatus,A     ; Execution result is stored in variable
                                ; (CmdStatus = 0: normal termination, other than 0: abnormal
                                ; termination)

;----------------------------
;END
;----------------------------
```

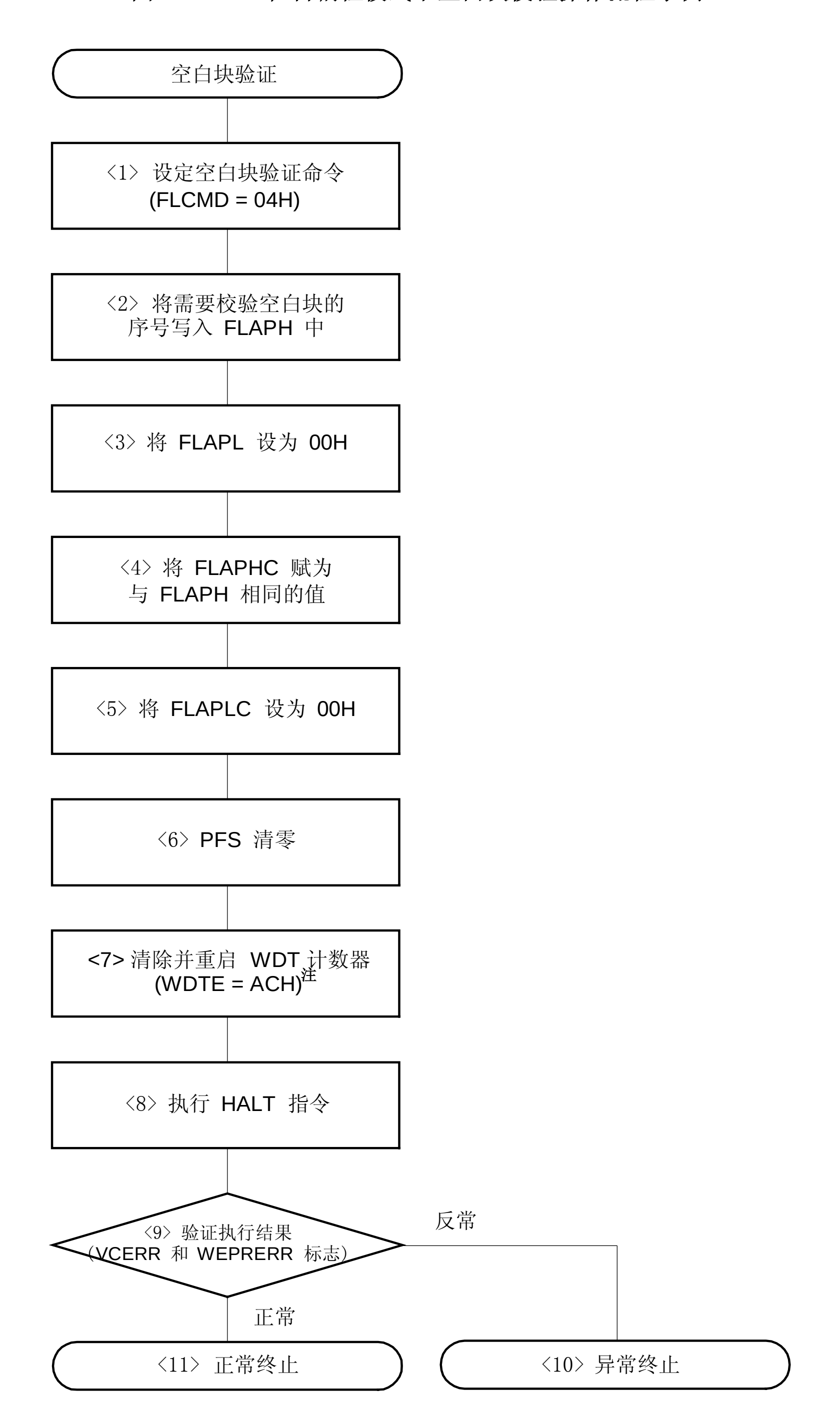
19.8.7　自编程模式下空白块的校验操作示例

自编程模式下空白块的校验操作过程如下：

〈1〉　设置编程命令寄存器(FLCMD)值为 04H（空白块校验）。
〈2〉　将需要执行空白块校验的块的序号放到 flash 地址指针 H 寄存器（FLAPH）中。
〈3〉　设置地址指针 L(FLAPL)为 00H。
〈4〉　将 FLAPH 的值写到到 flash 地址指针 H 比较寄存器(FLAPC)中。
〈5〉　设置 flash 地址指针 L 比较寄存器(FLAPLC)为 FFH。
〈6〉　清除 flash 状态寄存器（PFS）。
〈7〉　写 ACH 到看门狗定时器开放寄存器(WDTE)（清零并重启看门狗定时计数器）[注]。
〈8〉　执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
〈9〉　用 PFS 的第 1 位(VCERR)和第 2 位(WEPRERR)检查自编程是否产生错误。
　　错误 → 〈10〉
　　正常 → 〈11〉
〈10〉空白块校验非正常结束。
〈11〉空白块校验正常结束。

注　当看门狗定时器不使用时无需对其进行设置。

图 19-23. 在自编程模式下空白块校验操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 19-23 中的〈1〉到〈11〉 对应于 19.8.7 中的〈1〉到〈11〉 (前页)。

以下自编程模式下空白块校验程序示例。

```
;----------------------------
;START
;----------------------------

FlashBlockBlankCheck:
        MOV     FLCMD,#04H      ; Sets flash control command (block blank check)
        MOV     FLAPH,#07H      ; Sets number of block for blank check (block 7 is specified
                                ; here)
        MOV     FLAPL,#00H      ; Fixes FLAPL to “00H”
        MOV     FLAPHC,#07H     ; Sets blank check block compare number (same value as that of
                                ; FLAPH)
        MOV     FLAPLC,#0FFH    ; Fixes FLAPLC to “FFH”

        MOV     PFS,#00H        ; Clears flash status register
        MOV     WDTE,#0ACH      ; Clears & restarts WDT
        HALT                    ; Self programming is started
        MOV     A,PFS
        MOV     CmdStatus,A     ; Execution result is stored in variable
                                ; (CmdStatus = 0: normal termination, other than 0: abnormal
                                ; termination)

;----------------------------
;END
;----------------------------
```

19.8.8　自编程模式下字节写入操作示例

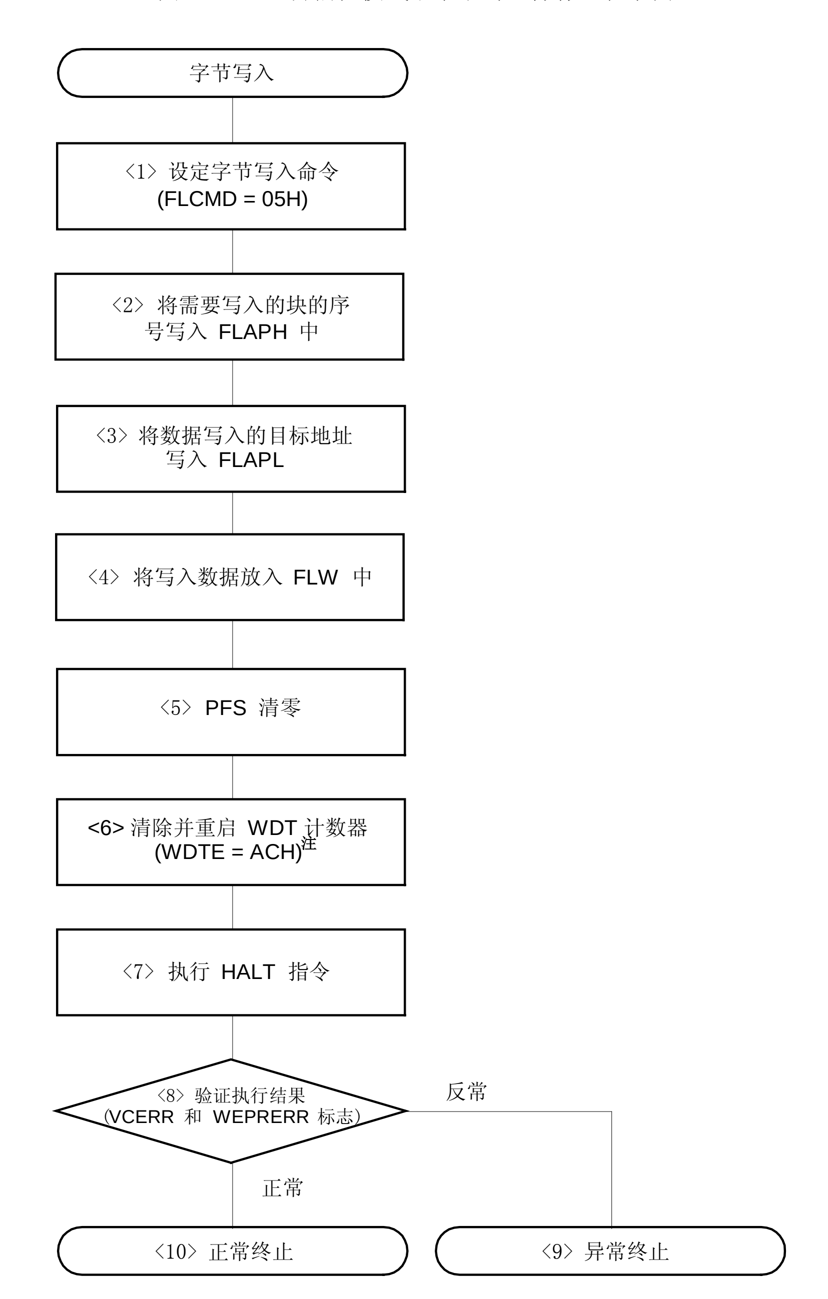
自编程模式下字节写入操作过程如下：

〈1〉　设置编程命令寄存器(FLCMD)值为 05H（字节写入）。
〈2〉　将需要被写入数据的块的序号放到 flash 地址指针 H 寄存器（FLAPH）中。
〈3〉　将需要被写入数据的地址放到 flash 地址指针 L 寄存器（FLAPL）中。
〈4〉　将需要写入的数据放到 flash 写入缓存寄存器（FLW）中。
〈5〉　清除 flash 状态寄存器（PFS）。
〈6〉　写 ACH 到看门狗定时器开放寄存器(WDTE)（清零并重启看门狗定时计数器）[注]。
〈7〉　执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
〈8〉　用 PFS 的第 1 位(VCERR)和第 2 位(WEPRERR)检查自编程是否产生错误。
　　　错误 → 〈9〉
　　　正常 → 〈10〉
〈9〉　字节写入进程非正常结束。
〈10〉字节写入进程正常结束。

注　　当看门狗定时器不使用时无需对其进行设置。

注意事项　　**如果写入失败，应先擦除块内容并再次写入。**

图 19-24. 自编程模式下字节写入操作流程示例



注 当看门狗定时器不使用时无需对其进行设置。

备注 图 19-24 中的〈1〉到〈10〉对应于 19.8.8 中的〈1〉到〈10〉（前页）。

以下是自编程模式下字节写入程序示例。

```
;----------------------------
;START
;----------------------------
FlashWrite:
        MOV     FLCMD,#05H      ; Sets flash control command (byte write)
        MOV     FLAPH,#07H      ; Sets address to which data is to be written, with
                                ; FLAPH (block 7 is specified here)
        MOV     FLAPL,#20H      ; Sets address to which data is to be written, with
                                ; FLAPL (address 20H is specified here)
        MOV     FLW,#10H        ; Sets data to be written (10H is specified here)

        MOV     PFS,#00H        ; Clears flash status register
        MOV     WDTE,#0ACH      ; Clears & restarts WDT
        HALT                    ; Self programming is started
        MOV     A,PFS
        MOV     CmdStatus,A     ; Execution result is stored in variable
                                ; (CmdStatus = 0: normal termination, other than 0: abnormal
                                ; termination)

;----------------------------
;END
;----------------------------
```

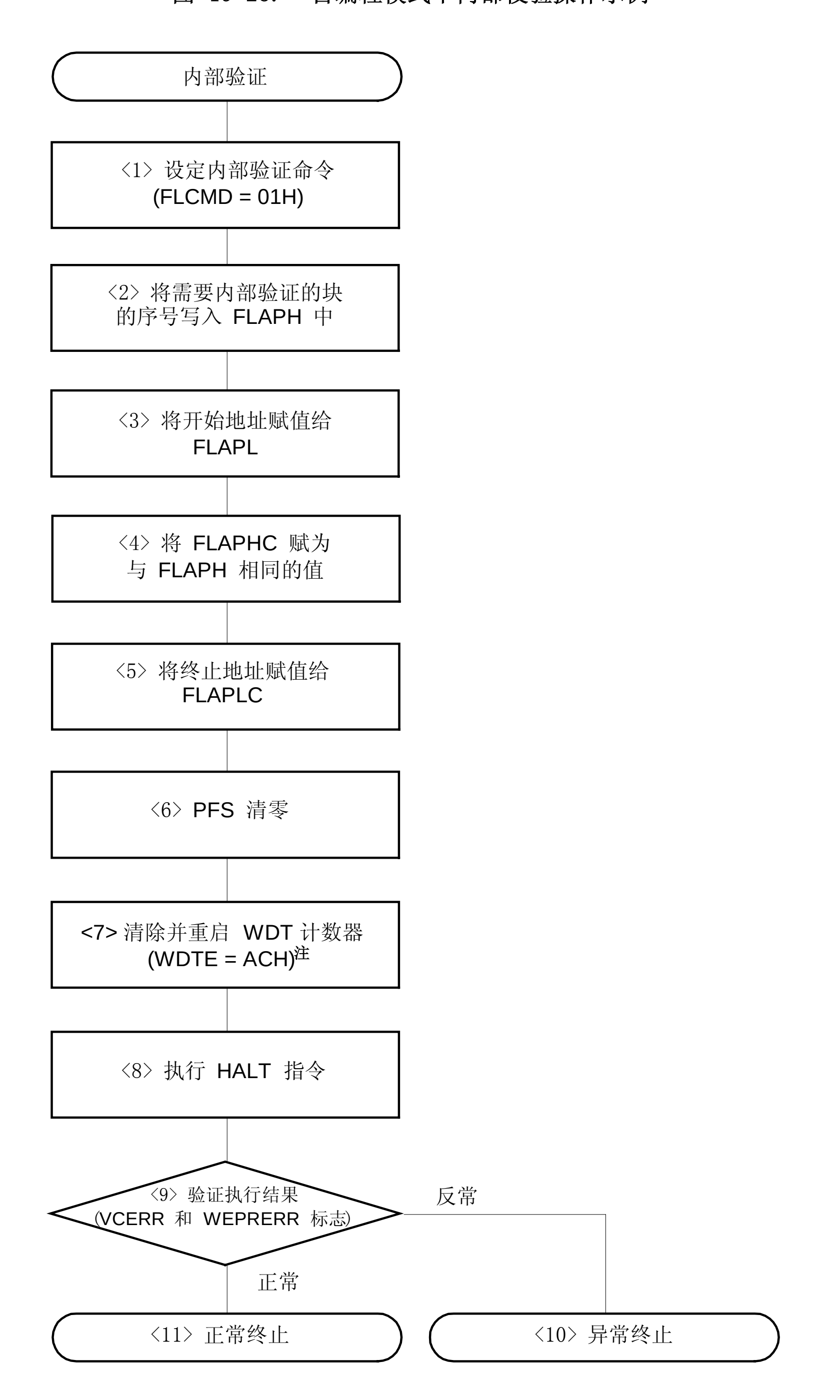
19.8.9　自编程模式下内部校验操作示例

自编程模式下内部校验操作过程如下：

〈1〉　设置编程命令寄存器(FLCMD)值为 01H（内部校验）。
〈2〉　将需要执行内部校验的块的序号值放到到 flash 地址指针 H(FLAPH)中。
〈3〉　设置内部校验的起始地址到 flash 地址指针 L(FLAPL)中。
〈4〉　将 FLAPH 的值写到到地址指针 H 比较寄存器(FLAPC)中。
〈5〉　设置内部校验的结束地址到 flash 地址指针 L 比较寄存器(FLAPLC)中。
〈6〉　清除 flash 状态寄存器（PFS）。
〈7〉　写 ACH 到看门狗定时器开放寄存器(WDTE)（清 0 并重启看门狗定时计数器）[注]。
〈8〉　执行 HALT 指令然后开始自编程（如果自编程已开始执行，则在 HALT 指令后立刻执行一条指令）。
〈9〉　用 PFS 的第 1 位(VCERR)和第 2 位(WEPRERR)检查自编程是否产生错误。
　　　错误 → 〈10〉
　　　正常 → 〈11〉
〈10〉内部校验非正常结束。
〈11〉内部校验正常结束。

注　当看门狗定时器不使用时无需对其进行设置。

图 19-25. 自编程模式下内部校验操作示例



注　　当看门狗定时器不使用时无需对其进行设置。

备注　　图 19-25 中的〈1〉到〈11〉 对应于 19.8.9 中的〈1〉到〈11〉 （前页）。

自编程模式下内部校验操作程序示例如下。

```
;----------------------------
;START
;----------------------------
FlashVerify:
        MOV       FLCMD,#01H        ; Sets flash control command (internal verify)
        MOV       FLAPH,#07H        ; Sets verify start address with FLAPH (block 7 is specified
                                    ; here)
        MOV       FLAPL,#00H        ; Sets verify start address with FLAPL (Address 00H is
                                    ; specified here)
        MOV       FLAPHC,#07H
        MOV       FLAPLC,#20H       ; Sets verify end address

        MOV       PFS,#00H          ; Clears flash status register
        MOV       WDTE,#0ACH        ; Clears & restarts WDT
        HALT                        ; Self programming is started
        MOV       A,PFS
        MOV       CmdStatus,A       ; Execution result is stored in variable
                                    ; (CmdStatus = 0: normal termination, other than 0: abnormal
                                    ; termination)

;----------------------------
;END
;----------------------------
```

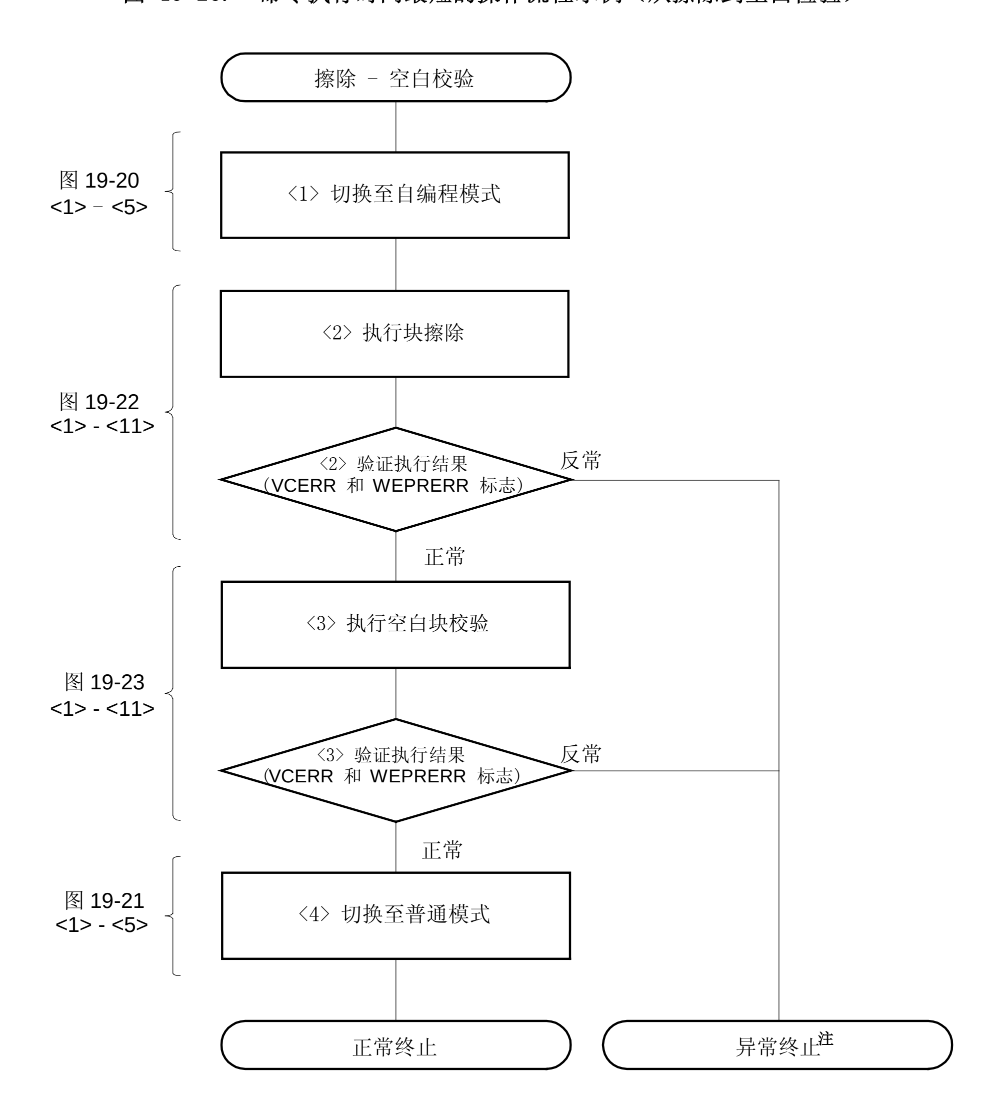
19.8.10 自编程模式下命令执行时间最短的操作示例

自编程模式下命令执行时间最短的操作过程如下：

(1) 从擦除到空白检验

〈1〉 将工作模式由普通模式切换至自编程模式(19.8.4 中的〈1〉 到 〈5〉)。
〈2〉 执行块擦除命令——错误检查 (19.8.6 中的〈1〉 到 〈11〉)。
〈3〉 执行空白块检验——错误检查(19.8.7 中的〈1〉 到 〈11〉)。
〈4〉 将工作模式由自编程模式切换回普通模式(19.8.5 中的〈1〉 到 〈5〉)。

图 19-26. 命令执行时间最短的操作流程示例（从擦除到空白检验）



注 为了让程序返回到正常的处理中需将工作模式切换回普通模式。

备注 图 19-26 中〈1〉到〈4〉对应于 19.8.10 (1) 中的〈1〉到〈4〉。

以下是自编程模式下命令执行时间最短的操作程序示例：

```
;----------------------------------------------------------------------
;START
;----------------------------------------------------------------------
        MOV     MK0,#11111111B      ; Masks all interrupts
        MOV     MK1,#11111111B

        DI

ModeOnLoop:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H         ; PFCMD register control
        MOV     FLPMC,#01H          ; FLPMC register control (sets value)
        MOV     FLPMC,#0FEH         ; FLPMC register control (inverts set value)
        MOV     FLPMC,#01H          ; Sets self programming mode with FLPMC register control (sets
                                    ; value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOnLoop         ; Checks completion of write to specific registers
                                    ; Repeats the same processing when an error occurs

FlashBlockErase:
        MOV     FLCMD,#03H          ; Sets flash control command (block erase)
        MOV     FLAPH,#07H          ; Sets number of block to be erased (block 7 is specified
                                    ; here)
        MOV     FLAPL,#00H          ; Fixes FLAPL to “00H”
        MOV     FLAPHC,#07H         ; Sets erase block compare number (same value as that of
                                    ; FLAPH)
        MOV     FLAPLC,#00H         ; Fixes FLAPLC to “00H”

        MOV     PFS,#00H            ; Clears flash status register
        MOV     WDTE,#0ACH          ; Clears & restarts WDT
        HALT                        ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError        ; Checks erase error
                                    ; Performs abnormal termination processing when an error
                                    ; occurs.

FlashBlockBlankCheck:
        MOV     FLCMD,#04H          ; Sets flash control command (block blank check)
        MOV     FLAPH,#07H          ; Sets number of block for blank check (block 7 is specified
                                    ; here)
        MOV     FLAPL,#00H          ; Fixes FLAPL to “00H”
```

```
        MOV     FLAPHC,#07H         ; Sets blank check block compare number (same value as of
                ; FLAPH)
        MOV     FLAPLC,#0FFH        ; Fixes FLAPLC to "FFH"
        MOV     PFS,#00H            ; Clears flash status register
        MOV     WDTE,#0ACH          ; Clears & restarts WDT
        HALT                        ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError        ; Checks blank check error
                                    ; Performs abnormal termination processing when an error
                ; occurs.

ModeOffLoop:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H         ; PFCMD register control
        MOV     FLPMC,#00H          ; FLPMC register control (sets value)
        MOV     FLPMC,#0FFH         ; FLPMC register control (inverts set value)
        MOV     FLPMC,#00H          ; Sets normal mode via FLPMC register control (sets value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOffLoop        ; Checks completion of write to specific registers
                                    ; Repeats the same processing when an error occurs

        MOV     MK0,#INT_MK0        ; Restores interrupt mask flag
        MOV     MK1,#INT_MK1

        EI

        BR      StatusNormal

;-----------------------------------------------------------------------
;END (abnormal termination processing); Perform processing to shift to
      normal mode in order to return to normal processing
;-----------------------------------------------------------------------
StatusError:

;-----------------------------------------------------------------------
;END (normal termination processing)
;-----------------------------------------------------------------------
StatusNormal:
```

(2)　从写操作到内部校验

〈1〉　将工作模式由普通模式切换至自编程模式（19.8.4 中的〈1〉到〈5〉）。

〈2〉　确定需要写入的数据。

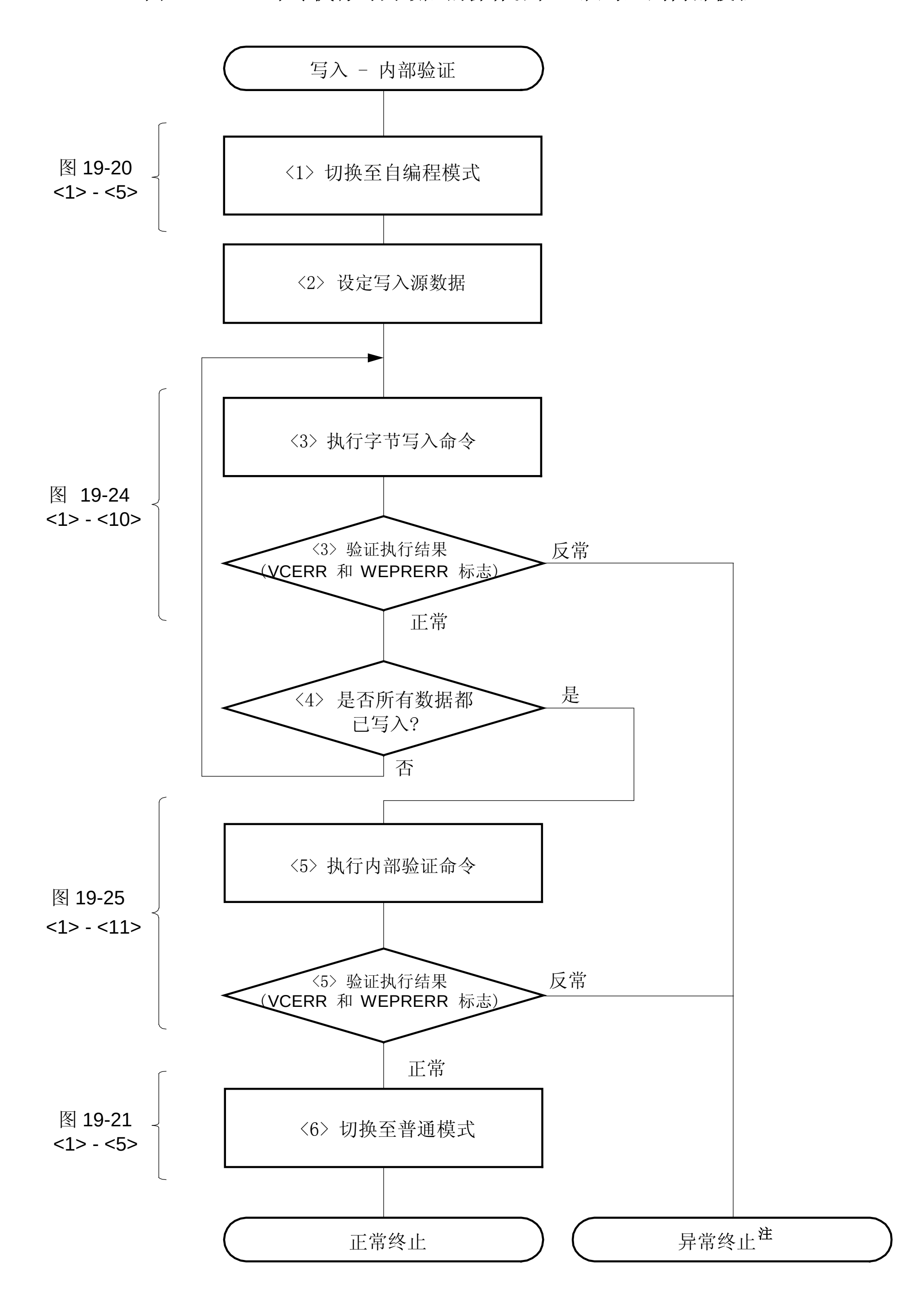
〈3〉　执行字节写入——错误检查（19.8.8 中的〈1〉到〈10〉）。

〈4〉　重复操作〈3〉直到将所有数据写入。

〈5〉　执行内部校验指令——错误检查（19.8.9 中的〈1〉到〈11〉）。

〈6〉　将工作模式由自编程模式切换回普通模式（19.8.5 中的〈1〉到〈5〉）。

图 19-27.　命令执行时间最短的操作流程（从写入到内部校验）



注　　为了让程序返回到正常的处理中需将工作模式切换回普通模式。

备注 图 19-27 中<1>到<6>对应于 19.8.10 (2) 中的<1>到<6>。

以下是自编程模式下命令执行时间最短（从写入到内部校验）的操作程序示例：

```
;----------------------------------------------------------------------
;START
;----------------------------------------------------------------------
        MOV       MK0,#11111111B      ; Masks all interrupts
        MOV       MK1,#11111111B

        DI

ModeOnLoop:
        MOV       PFS,#00H
        MOV       PFCMD,#0A5H         ; PFCMD register control
        MOV       FLPMC,#01H          ; FLPMC register control (sets value)
        MOV       FLPMC,#0FEH         ; FLPMC register control (inverts set value)
        MOV       FLPMC,#01H          ; Sets self programming mode with FLPMC register control
                                      ; (sets value)

        MOV       A,PFS
        CMP       A,#00H
        BNZ       $ModeOnLoop         ; Checks completion of write to specific registers
                                      ; Repeats the same processing when an error occurs

FlashWrite:
        MOVW      HL,#DataAdrTop      ; Sets address at which data to be written is located
        MOVW      DE,#WriteAdr        ; Sets address at which data is to be written
FlashWriteLoop:
        MOV       FLCMD,#05H          ; Sets flash control command (byte write)
        MOV       A,D
        MOV       FLAPH,A             ; Sets address at which data is to be written
        MOV       A,E
        MOV       FLAPL,A             ; Sets address at which data is to be written
        MOV       A,[HL]
        MOV       FLW,A               ; Sets data to be written

        MOV       PFS,#00H            ; Clears flash status register
        MOV       WDTE,#0ACH          ; Clears & restarts WDT
        HALT                          ; Self programming is started
        MOV       A,PFS
        CMP       A,#00H
        BNZ       $StatusError        ; Checks write error
                                      ; Performs abnormal termination processing when an error
                                      ; occurs.

        INCW      HL                  ; address at which data to be written is located + 1
```

```
        MOVW    AX,HL
        CMPW    AX,#DataAdrBtm      ; Performs internal verify processing
        BNC     $FlashVerify        ; if write of all data is completed

        INCW    DE                  ; Address at which data is to be written + 1
        BR      FlashWriteLoop

FlashVerify:
        MOVW    HL,#WriteAdr        ; Sets verify address

        MOV     FLCMD,#01H          ; Sets flash control command (internal verify)
        MOV     A,H
        MOV     FLAPH,A             ; Sets verify start address
        MOV     A,L
        MOV     FLAPL,A             ; Sets verify start address
        MOV     A,D
        MOV     FLAPHC,A            ; Sets verify end address
        MOV     A,E
        MOV     FLAPLC,A            ; Sets verify end address

        MOV     PFS,#00H            ; Clears flash status register
        MOV     WDTE,#0ACH          ; Clears & restarts WDT
        HALT                        ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError        ; Checks internal verify error
                                    ; Performs abnormal termination processing when an error
                                    ; occurs.

ModeOffLoop:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H         ; PFCMD register control
        MOV     FLPMC,#00H          ; FLPMC register control (sets value)
        MOV     FLPMC,#0FFH         ; FLPMC register control (inverts set value)
        MOV     FLPMC,#00H          ; Sets normal mode via FLPMC register control (sets value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOffLoop        ; Checks completion of write to specific registers
                                    ; Repeats the same processing when an error occurs

        MOV     MK0,#INT_MK0        ; Restores interrupt mask flag
        MOV     MK1,#INT_MK1

        EI

        BR      StatusNormal
```

```
;-----------------------------------------------------------------------
;END (abnormal termination processing); Perform processing to shift to
      normal mode in order to return to normal processing
;-----------------------------------------------------------------------
StatusError:


;-----------------------------------------------------------------------
;END (normal termination processing)
;-----------------------------------------------------------------------
StatusNormal:


;-----------------------------------------------------------------------
; Data to be written
;-----------------------------------------------------------------------
DataAdrTop:
        DB        XXH
        DB        XXH
        DB        XXH
        DB        XXH

         :
         :

        DB        XXH
DataAdrBtm:
;-----------------------------------------------------------------------
```

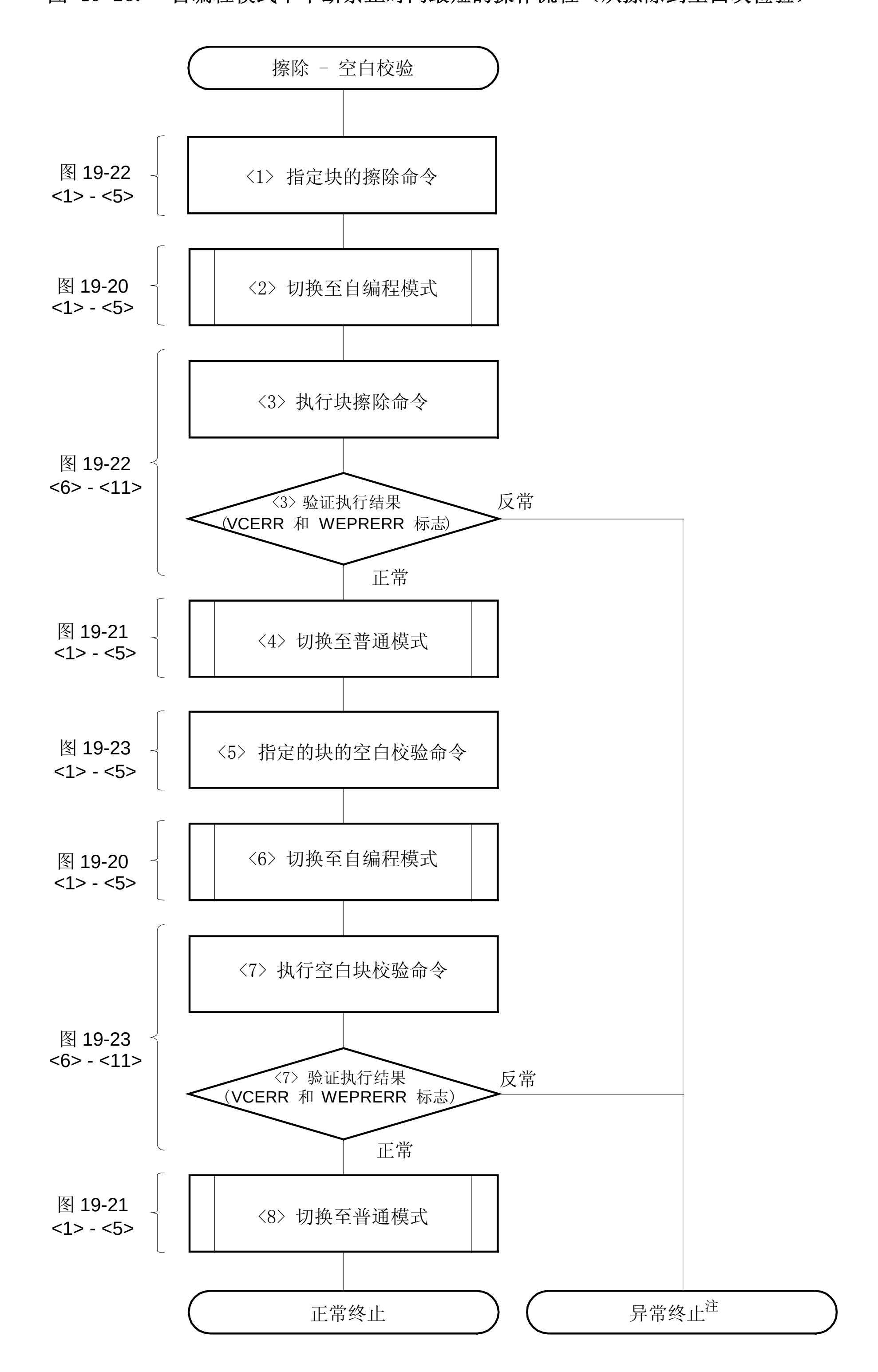
19.8.11 自编程模式下中断禁止时间最短的操作过程

自编程模式下中断禁止时间最短的操作过程如下：

(1)　从擦除到空白检验

〈1〉　设置块擦除命令（19.8.6 中的〈1〉 到 〈5〉）。
〈2〉　将工作模式由普通模式切换至自编程模式（19.8.4 中的〈1〉 到 〈5〉）。
〈3〉　执行块擦除命令——错误检查（19.8.6 中的〈6〉 到 〈11〉）。
〈4〉　将工作模式由自编程模式切换至普通模式（19.8.5 中的〈1〉 到 〈5〉）。
〈5〉　设置空白块检验命令（19.8.7 中的〈1〉 到 〈5〉）。
〈6〉　将工作模式由普通模式切换至自编程模式（19.8.4 中的〈1〉 到 〈5〉）。
〈7〉　执行空白块检验——错误检查（19.8.7 中的〈6〉 到 〈11〉）。
〈8〉　将工作模式由自编程模式切换回普通模式（19.8.5 中的〈1〉 到 〈5〉）。

图 19-28.　自编程模式下中断禁止时间最短的操作流程（从擦除到空白块检验）



注　　为了让程序返回到正常的处理中需将工作模式切换回普通模式。

备注　　图 19-28 中〈1〉到〈8〉对应于 19.8.11 （1） 中〈1〉到〈8〉(前页)。

以下是自编程模式下中断禁止（从擦除到空白检验）时间最短的程序示例：

```
;----------------------------------------------------------------------
;START
;----------------------------------------------------------------------
FlashBlockErase:
        ; Sets erase command
        MOV     FLCMD,#03H       ; Sets flash control command (block erase)
        MOV     FLAPH,#07H       ; Sets number of block to be erased (block 7 is specified here)
        MOV     FLAPL,#00H       ; Fixes FLAPL to “00H”
        MOV     FLAPHC,#07H      ; Sets erase block compare number (same value as that of FLAPH)
        MOV     FLAPLC,#00H      ; Fixes FLAPLC to “00H”

        CALL    !ModeOn          ; Shift to self programming mode

        ; Execution of erase command
        MOV     PFS,#00H         ; Clears flash status register
        MOV     WDTE,#0ACH       ; Clears & restarts WDT
        HALT                     ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError     ; Checks erase error
                                 ; Performs abnormal termination processing when an error
                                 ; occurs.

        CALL    !ModeOff         ; Shift to normal mode


        ; Sets blank check command
        MOV     FLCMD,#04H       ; Sets flash control command (block blank check)
        MOV     FLAPH,#07H       ; Sets block number for blank check (block 7 is specified here)
        MOV     FLAPL,#00H       ; Fixes FLAPL to “00H”
        MOV     FLAPHC,#07H      ; Sets blank check block compare number (same value as that of
                                 ; FLAPH)
        MOV     FLAPLC,#0FFH     ; Fixes FLAPLC to “FFH”

        CALL    !ModeOn          ; Shift to self programming mode

        ; Execution of blank check command
        MOV     PFS,#00H         ; Clears flash status register
        MOV     WDTE,#0ACH       ; Clears & restarts WDT
        HALT                     ; Self programming is started
        MOV     A,PFS
        CMP     A,#00H
        BNZ     $StatusError     ; Checks blank check error
                                 ; Performs abnormal termination processing when an error occurs

        CALL    !ModeOff         ; Shift to normal mode
```

```
        BR          StatusNormal

;---------------------------------------------------------------------
;END (abnormal termination processing); Perform processing to shift to
      normal mode in order to return to normal processing
;---------------------------------------------------------------------
StatusError:



;---------------------------------------------------------------------
;END (normal termination processing)
;---------------------------------------------------------------------
StatusNormal:



;---------------------------------------------------------------------
;Processing to shift to self programming mode
;---------------------------------------------------------------------
ModeOn:
        MOV         MK0,#11111111B       ; Masks all interrupts
        MOV         MK1,#11111111B

        DI

ModeOnLoop:
        MOV         PFS,#00H
        MOV         PFCMD,#0A5H        ; PFCMD register control
        MOV         FLPMC,#01H         ; FLPMC register control (sets value)
        MOV         FLPMC,#0FEH        ; FLPMC register control (inverts set value)
        MOV         FLPMC,#01H         ; Sets self programming mode via FLPMC register control (sets
                                       ; value)

        MOV         A,PFS
        CMP         A,#00H
        BNZ         $ModeOnLoop        ; Checks completion of write to specific registers
                                       ; Repeats the same processing when an error occurs

        RET

;---------------------------------------------------------------------
; Processing to shift to normal mode
;---------------------------------------------------------------------
ModeOff:
        MOV         PFS,#00H
        MOV         PFCMD,#0A5H        ; PFCMD register control
        MOV         FLPMC,#00H         ; FLPMC register control (sets value)
        MOV         FLPMC,#0FFH        ; FLPMC register control (inverts set value)
```

```
MOV     FLPMC,#00H      ; Sets normal mode via FLPMC register control (sets value)

MOV     A,PFS
CMP     A,#00H
BNZ     $ModeOff        ; Checks completion of write to specific registers
                        ; Repeats the same processing when an error occurs。

MOV     MK0,#INT_MK0    ; Restores interrupt mask flag
MOV     MK1,#INT_MK1

EI

RET
```

(2) 从写操作到内部校验

〈1〉 设置需要写入的数据。

〈2〉 设置字节写入命令(19.8.8 中的〈1〉 到 〈4〉)。

〈3〉 将工作模式由普通模式切换到自编程模式(19.8.4 中的〈1〉 到 〈5〉)。

〈4〉 执行字节写入命令——错误检查(19.8.8 中的〈5〉到〈10〉)。

〈5〉 将工作模式由自编程模式切换回普通模式(19.8.5 中的〈1〉 到 〈5〉)。

〈6〉 重复操作〈2〉到〈5〉直到将所有数据写入。

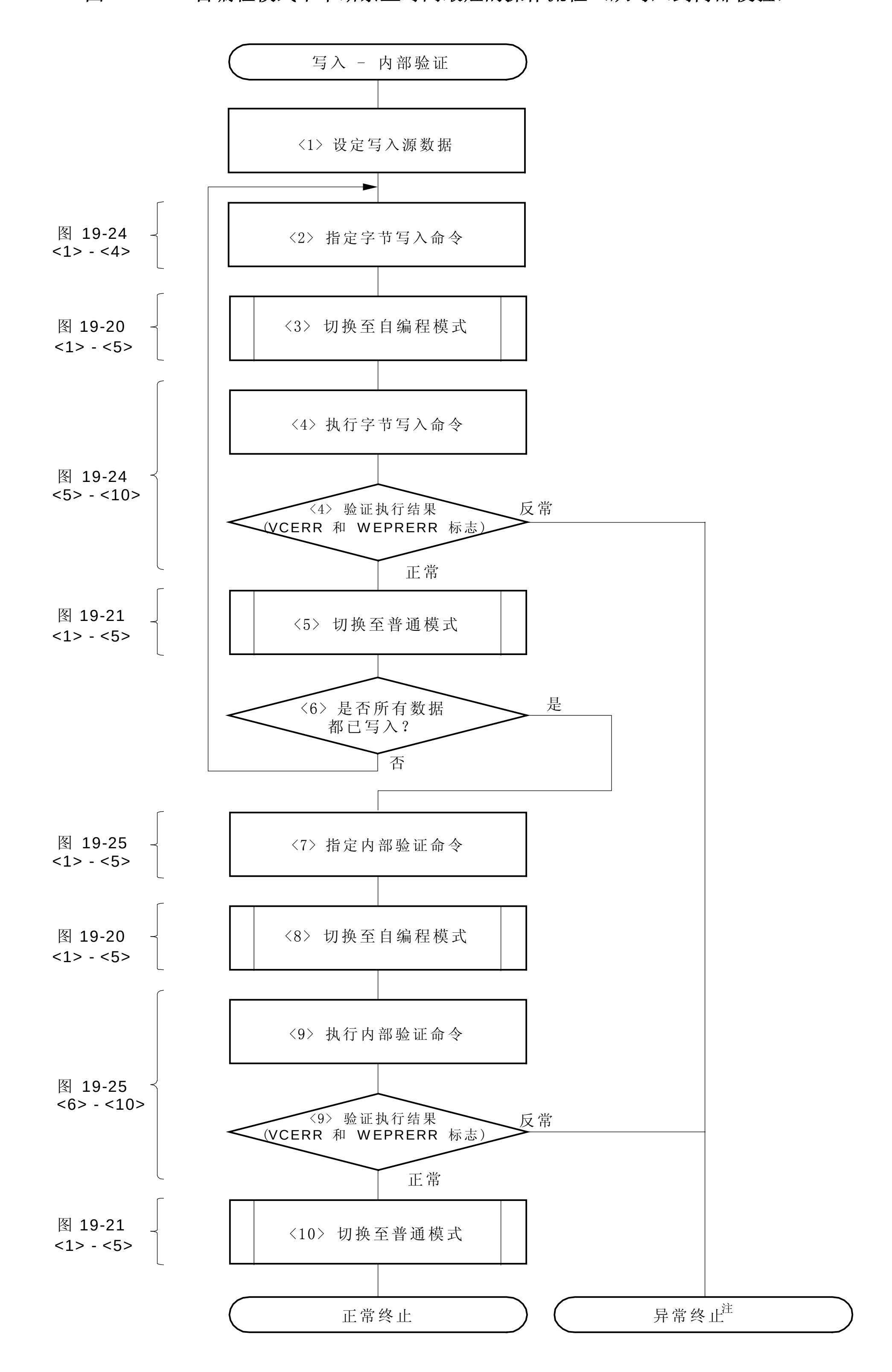
〈7〉 设定内部校验命令(19.8.9 中的〈1〉 到 〈5〉)。

〈8〉 将工作模式由普通模式切换到自编程模式(19.8.4 中的〈1〉 到 〈5〉)。

〈9〉 执行内部校验指令——错误检查(19.8.9 中的〈6〉 到 〈11〉)。

〈10〉 将工作模式由自编程模式切换回普通模式(19.8.5 中的〈1〉 到 〈5〉)。

图 19-29.　自编程模式下中断禁止时间最短的操作流程（从写入到内部校验）



注　为了让程序返回到正常的处理中需将工作模式切换回普通模式。

备注　图 19-29 中〈1〉到〈10〉是相应于 19.8.11（2）中的〈1〉到〈10〉(前页)。

以下是自编程模式下中断禁止（从写入到内部校验）时间最短的操作程序示：

```
;----------------------------------------------------------------------
;START
;----------------------------------------------------------------------
        ; Sets write command
FlashWrite:
        MOVW      HL,#DataAdrTop     ; Sets address at which data to be written is located
        MOVW      DE,#WriteAdr       ; Sets address at which data is to be written

FlashWriteLoop:
        MOV       FLCMD,#05H         ; Sets flash control command (byte write)
        MOV       A,D
        MOV       FLAPH,A            ; Sets address at which data is to be written
        MOV       A,E
        MOV       FLAPL,A            ; Sets address at which data is to be written
        MOV       A,[HL]
        MOV       FLW,A              ; Sets data to be written

        CALL      !ModeOn            ; Shift to self programming mode

        ; Execution of write command
        MOV       PFS,#00H           ; Clears flash status register
        MOV       WDTE,#0ACH         ; Clears & restarts WDT
        HALT                         ; Self programming is started
        MOV       A,PFS
        CMP       A,#00H
        BNZ       $StatusError       ; Checks write error
                                     ; Performs abnormal termination processing when an error
                                     ; occurs.

        CALL      !ModeOff           ; Shift to normal mode

        MOV       MK0,#INT_MK0       ; Restores interrupt mask flag
        MOV       MK1,#INT_MK1

        EI

        ; Judgment of writing all data
        INCW      HL                 ; Address at which data to be written is located + 1
        MOVW      AX,HL
        CMPW      AX,#DataAdrBtm     ; Performs internal verify processing
        BNC       $FlashVerify       ; if write of all data is completed

        INCW      DE                 ; Address at which data is to be written + 1
        BR        FlashWriteLoop

        ; Setting internal verify command
```

```
FlashVerify:
        MOVW      HL,#WriteAdr       ; Sets verify address

        MOV       FLCMD,#01H         ; Sets flash control command (internal verify)
        MOV       A,H
        MOV       FLAPH,A            ; Sets verify start address
        MOV       A,L
        MOV       FLAPL,A            ; Sets verify start address
        MOV       A,D
        MOV       FLAPHC,A           ; Sets verify end address
        MOV       A,E
        MOV       FLAPLC,A           ; Sets verify end address

        CALL      !ModeOn            ; Shift to self programming mode

        ; Execution of internal verify command
        MOV       PFS,#00H           ; Clears flash status register
        MOV       WDTE,#0ACH         ; Clears & restarts WDT
        HALT                         ; Self programming is started
        MOV       A,PFS
        CMP       A,#00H
        BNZ       $StatusError       ; Checks internal verify error
                                     ; Performs abnormal termination processing when an error occurs

        CALL      !ModeOff           ; Shift to normal mode

        BR        StatusNormal

;----------------------------------------------------------------------
;END (abnormal termination processing); Perform processing to shift to
      normal mode in order to return to normal processing
;----------------------------------------------------------------------
StatusError:



;----------------------------------------------------------------------
;END (normal termination processing)
;----------------------------------------------------------------------
StatusNormal:



;----------------------------------------------------------------------
;Processing to shift to self programming mode
;----------------------------------------------------------------------
ModeOn:
        MOV       MK0,#11111111B     ; Masks all interrupts
        MOV       MK1,#11111111B
```

```
        DI

ModeOnLoop:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H     ; PFCMD register control
        MOV     FLPMC,#01H      ; FLPMC register control (sets value)
        MOV     FLPMC,#0FEH     ; FLPMC register control (inverts set value)
        MOV     FLPMC,#01H      ; Sets self programming mode via FLPMC register control (sets
                                ; value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOnLoop     ; Checks completion of write to specific registers
                                ; Repeats the same processing when an error occurs

        RET


;-----------------------------------------------------------------------
; Processing to shift to normal mode
;-----------------------------------------------------------------------
ModeOff:
        MOV     PFS,#00H
        MOV     PFCMD,#0A5H     ; PFCMD register control
        MOV     FLPMC,#00H      ; FLPMC register control (sets value)
        MOV     FLPMC,#0FFH     ; FLPMC register control (inverts set value)
        MOV     FLPMC,#00H      ; Sets normal mode via FLPMC register control (sets value)

        MOV     A,PFS
        CMP     A,#00H
        BNZ     $ModeOff        ; Checks completion of write to specific registers
                                ; Repeats the same processing when an error occurs。

        MOV     MK0,#INT_MK0    ; Restores interrupt mask flag
        MOV     MK1,#INT_MK1

        EI

        RET


;-----------------------------------------------------------------------
;Data to be written
;-----------------------------------------------------------------------
DataAdrTop:
        DB      XXH
        DB      XXH
        DB      XXH
```

```
        DB          XXH

        :
        :

        DB          XXH
DataAdrBtm:
;-------------------------------------------------------------------
```

第二十章　指令集

本章列出了 78K0S/KB1+ 的指令集。如需了解每种指令的操作和机器语言(操作代码),可参考 **78K/0S 系列指令用户手册** (U11047E)。

20.1　操作

20.1.1 操作数标识符和描述方法

根据指令操作数描述方法 (详情可参见汇编程序规范),在每种指令的“操作数”栏列出操作数。 如果有两种或两种以上的表示方法, 可任选其一。 大写字母和符号 #, !, $, 和 [ ] 是关键字,必须按其原样书写。 每种符号的含义如下所示。

- #:　立即数标识
- !:　绝对地址标识
- $:　相对地址标识
- [ ]: 间接地址标识

立即数用来描述一个数值型数据或标号。 当使用标志时,注意必须加上符号#, !, $ 和[ ] 。

对于操作数寄存器标识符 r 和 rp,功能名称(X, A, C 等)或绝对名称(下表括号中的名称, R0, R1, R2 等) 都可用于标识。

表 20-1. 操作数标识符和标识方法

标识符	描述方法
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) 专用寄存器符号
saddr saddrp	FE20H ～ FF1FH　立即数或标号 FE20H ～ FF1FH　立即数或标号(仅用于偶地址)
addr16 addr5	0000H ～ FFFFH 立即数或标号(仅用于 16 位数据传输指令的偶地址) 0040H ～ 007FH 立即数或标号(仅用于偶地址)
word byte bit	16 位立即数或标号 8 位立即数或标号 3 位立即数或标号

备注　专用寄存器符号参见**表 3-3　特殊功能寄存器**。

20.1.2　“操作”栏的描述

A:　A 寄存器; 8 位 累加器
X:　X 寄存器
B:　B 寄存器
C:　C 寄存器
D:　D 寄存器
E:　E 寄存器
H:　H 寄存器
L:　L 寄存器
AX:　AX 寄存器对; 16 位累加器
BC:　BC 寄存器对
DE:　DE 寄存器对
HL:　HL 寄存器对
PC:　程序计数器
SP:　堆栈指针
PSW:　程序状态字
CY:　进位标志
AC:　辅助进位标志
Z:　零标志
IE:　中断请求允许标志
( ):　括号中的地址或寄存器所指的存储单元内容
×H, ×L:　16 位寄存器的高 8 位和低 8 位
∧:　逻辑与(AND)
∨:　逻辑或(OR)
∀:　逻辑异或
—:　数据取反
addr16:　16 位立即数或标号
jdisp8:　带符号的 8 位数据 (偏移量)

20.1.3　“标志”栏的描述

(空):　不受影响
0:　清零
1:　设置为 1
×:　根据结果置位/清零
R:　恢复先前保存的值

20.2 操作列表

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
MOV	r, #byte	3	6	r ← byte			
	saddr, #byte	3	6	(saddr) ← byte			
	sfr, #byte	3	6	sfr ← byte			
	A, r 注1	2	4	A ← r			
	r, A 注1	2	4	r ← A			
	A, saddr	2	4	A ← (saddr)			
	saddr, A	2	4	(saddr) ← A			
	A, sfr	2	4	A ← sfr			
	sfr, A	2	4	sfr ← A			
	A, !addr16	3	8	A ← (addr16)			
	!addr16, A	3	8	(addr16) ← A			
	PSW, #byte	3	6	PSW ← byte	×	×	×
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	×	×	×
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL + byte]	2	6	A ← (HL + byte)			
	[HL + byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r 注2	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ sfr			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [HL, byte]	2	8	A ↔ (HL + byte)			

注 1. “r = A”除外
2. “r = A, X”除外

备注 一个指令周期是指由处理器时钟控制寄存器(PCC)选择的 CPU 时钟(f_{CPU})。

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp 注	1	4	AX ← rp			
	rp, AX 注	1	4	rp ← AX			
XCHW	AX, rp 注	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A − byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) − byte	×	×	×
	A, r	2	4	A, CY ← A − r	×	×	×
	A, saddr	2	4	A, CY ← A − (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A − (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A − (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A − (HL + byte)	×	×	×

注 仅当 rp = BC, DE 或 HL 时。

备注 一个指令周期是指由处理器时钟控制寄存器(PCC)选择的 CPU 时钟(f_{CPU})。

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A⊻ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ⊻ byte	×		
	A, r	2	4	A ← A⊻ r	×		
	A, saddr	2	4	A ← A⊻ (saddr)	×		
	A, !addr16	3	8	A ← A⊻ (addr16)	×		
	A, [HL]	1	6	A ← A⊻ (HL)	×		
	A, [HL + byte]	2	6	A ← A⊻ (HL + byte)	×		

备注 一个指令周期是指由处理器时钟控制寄存器(PCC)选择的 CPU 时钟(f_{CPU})。

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
CMP	A, #byte	2	4	A − byte	×	×	×
	saddr, #byte	3	6	(saddr) − byte	×	×	×
	A, r	2	4	A − r	×	×	×
	A, saddr	2	4	A − (saddr)	×	×	×
	A, !addr16	3	8	A − (addr16)	×	×	×
	A, [HL]	1	6	A − (HL)	×	×	×
	A, [HL + byte]	2	6	A − (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX − word	×	×	×
CMPW	AX, #word	3	6	AX − word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r − 1	×	×	
	saddr	2	4	(saddr) ← (saddr) − 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp − 1			
ROR	A, 1	1	2	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

备注　一个指令周期是指由处理器时钟控制寄存器(PCC)选择的 CPU 时钟(f_{CPU})。

助记符	操作数	字节数	时钟	操作	标志		
					Z	AC	CY
CALL	!addr16	3	6	(SP − 1) ← $(PC + 3)_H$, (SP − 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP − 2			
CALLT	[addr5]	1	8	(SP − 1) ← $(PC + 1)_H$, (SP − 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP − 2			
RET		1	6	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
RETI		1	8	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
PUSH	PSW	1	2	(SP − 1) ← PSW, SP ← SP − 1			
	rp	1	4	(SP − 1) ← rp_H, (SP − 2) ← rp_L, SP ← SP − 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC_H ← A, PC_L ← X			
BC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	B ← B − 1, 然后 PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C − 1, 然后 PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) − 1, 然后 PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	无操作			
EI		3	6	IE ← 1（允许中断）			
DI		3	6	IE ← 0（禁止中断）			
HALT		1	2	设置 HALT 模式			
STOP		1	2	设置 STOP 模式			

备注 一个指令周期是指由处理器时钟控制寄存器(PCC)选择的 CPU 时钟(f_{CPU})。

20.3 按寻址类型列出指令

(1)　8 位指令

MOV，XCH，ADD，ADDC，SUB，SUBC，AND，OR，XOR，CMP，INC，DEC，ROR，ROL，RORC，ROLC，PUSH，POP，DBNZ

第 2 操作数 第 1 操作数	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte]	$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[注] XCH[注] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL + byte]		MOV											

注　　r = A 除外。

(2) 16 位指令

MOVW，XCHW，ADDW，SUBW，CMPW，PUSH，POP，INCW，DECW

第 2 操作数 / 第 1 操作数	#word	AX	rp 注	saddrp	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW 注				INCW DECW PUSH POP
saddrp		MOVW				
sp		MOVW				

注 仅当 rp = BC,DE 或 HL 时。

(3) 位操作指令

SET1，CLR1，NOT1，BT，BF

第 2 操作数 / 第 1 操作数	$addr16	无
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4)　调用指令/转移指令

CALL，CALLT，BR，BC，BNC，BZ，BNZ，DBNZ

第 2 操作数 / 第 1 操作数	AX	!addr16	[addr5]	$addr16
基本指令	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
复合指令				DBNZ

(5) 其他指令

RET，RETI，NOP，EI，DI，HALT，STOP

第二十一章　电气特性（目标值）

这些特性值仅仅是目标值，可能不满足大批量生产的产品。

最大额定值（T_A = 25℃）

参数	符号	条件	范围	单位
电源电压	V_{DD}		–0.3 ～ +6.5	V
	V_{SS}		–0.3 ～ +0.3	V
	AV_{REF}		–0.3 ～ V_{DD} + 0.3 [注]	V
	AV_{SS}		–0.3 ～ +0.3	V
输入电压	V_{I1}	P00 ～ P03，P30 ～ P34，P40 ～ P47，P120 ～ P123	–0.3 ～ V_{DD} + 0.3 [注]	V
	V_{I2}	P20 ～ P23	– 0.3 ～ AV_{REF} + 0.3 [注] and –0.3 ～ V_{DD} + 0.3 [注]	V
输出电压	V_O		–0.3 ～ V_{DD} + 0.3 [注]	V
模拟输入电压	V_{AN}		AV_{SS} – 0.3 ～ AV_{REF} + 0.3 [注] and –0.3 ～ V_{DD} + 0.3 [注]	V
输出电流,高	I_{OH}	每个引脚	–10	mA
		所有引脚	–44	mA
输出电流,低	I_{OL}	每个引脚	20	mA
		所有引脚	44	mA
工作环境温度	T_A	在通常操作模式下	–40 ～ +85	℃
		在 Flash 存储器编程期间		
存储温度	T_{stg}		–40 ～ +125	℃

注　　必须为 6.5V 或更低。

注意事项　任何一项参数哪怕是在瞬间超过最大额定值,都会使产品质量受到影响. 也就是说，最大额定值是产品濒临物理损坏的临界点，因此,必须保证产品在不超过最大额定值的条件下使用。

备注　　除非有另外的规定，复用功能引脚的特性与端口引脚的相同。

X1 振荡器特性（T_A = −40 ～ +85℃, V_{DD} = 2.0 ～ 5.5 V[注 1], V_{SS} = 0 V）

振荡器	建议使用的电路	参数	条件	最小	典型	最大	单位
陶瓷振荡器	V_{SS} X1 X2 C1 C2	振荡频率 (f_X)[注 2]		1		10.0	MHz
晶体振荡器	V_{SS} X1 X2 C1 C2	振荡频率(f_X)[注 2]		1		10.0	MHz
外部时钟	X1	X1 输入频率(f_X)[注 2]	2.7 V ≤ V_{DD} ≤ 5.5 V	1		10.0	MHz
			2.0 V ≤ V_{DD} < 2.7 V	1		5.0	
		X1 输入高/低电平宽度 (t_{XH}, t_{XL})	2.7 V ≤ V_{DD} ≤ 5.5 V	0.045		0.5	μs
			2.0 V ≤ V_{DD} < 2.7 V	0.09		0.5	

注　1.　该产品的电压使用范围为 2.2 ～ 5.5 V ,因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

　　　2.　仅表示振荡器的特性，如需了解指令执行时间,可参见 AC 特性。

注意事项　在使用 X1 振荡器时，上表中的虚线框内部分的连线应遵从如下的连接方式，以防止线间电容产生不利影响。

- 连接线越短越好。
- 连接线不与其他信号线交叉。
- 如果信号线流经的电流变化较大,则不要在其周围连线。
- 要保持振荡器电容器的接地点电压与 V_{SS} 相同。
- 避免大电流从电容到地的连线上流过。
- 不要从振荡器获取信号。

备注　对于振荡器选择和振荡器常量,用户需要自己测试振荡器或要求制造商测试。

高速 Ring-OSC 振荡器特性 （T_A = −40 ～ +85°C, V_{DD} = 2.0 ～ 5.5 V[注1], f_X = 8 MHz[注2]）

振荡器	参数	条件		最小	典型	最大	单位
片内高速 Ring-OSC	振荡频率 (f_X)	2.7 V ≤ V_{DD} ≤ 5.5 V	T_A = −10 ～ +80°C			±3	%
			T_A = −40 ～ +85°C			±5	%
		2.0 V ≤ V_{DD} ＜ 2.7 V				T.B.D	%

注 1. 该产品的电压使用范围为 2.2 ～ 5.5 V，因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。
2. 仅表示振荡器的特性，如需了解指令执行时间,可参见 AC 特性。

低速 Ring-OSC 振荡器特性 （T_A = −40 ～ +85°C, V_{DD} = 2.0 ～ 5.5 V[注], V_{SS} = 0 V）

振荡器	参数	条件	最小	典型	最大	单位
片内低速 Ring-OSC	振荡频率 (f_{RL})		120	240	480	kHz

注 该产品的电压使用范围为 2.2 ～ 5.5 V，因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

DC 特性 (T_A = −40 ～ +85°C, V_{DD} = 2.0 ～ 5.5 V[注], V_{SS} = 0 V) (1/2)

参数	符号	条件			最小	典型	最大	单位
输出电流，高	I_{OH1}	除了 P20～P23 之外的引脚	每个引脚	2.0 V ≤ V_{DD} ≤ 5.5 V			−5	mA
			所有引脚	4.0 V ≤ V_{DD} ≤ 5.5 V			−25	mA
				2.0 V ≤ V_{DD} < 4.0 V			−15	mA
	I_{OH2}	P20 ～ P23	每个引脚	2.0 V ≤ AV_{REF} ≤ 5.5 V			−5	mA
			所有引脚	2.0 V ≤ AV_{REF} ≤ 5.5 V			−15	mA
输出电流，低	I_{OL}	每个引脚		2.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
		所有引脚		4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
				2.0 V ≤ V_{DD} < 4.0 V			15	mA
输入电压，高	V_{IH1}	P00 ～ P03，P30 ～ P34，P40 ～ P47，P120，P123			0.8V_{DD}		V_{DD}	V
	V_{IH2}	P20 ～ P23			0.7AV_{REF}		AV_{REF}	V
	V_{IH3}	P121，P122			0.8V_{DD}		V_{DD}	V
输入电压，低	V_{IL1}	P00 ～ P03，P30 ～ P34，P40 ～ P47，P120，P123			0		0.2V_{DD}	V
	V_{IL2}	P20 ～ P23			0		0.3AV_{REF}	V
	V_{IL3}	P121，P122			0		0.3V_{DD}	V
输出电压，高	V_{OH1}	除了 P20 至 P23 引脚之外的所有引脚 I_{OH1} = −15 mA		4.0 V ≤ V_{DD} ≤ 5.5 V I_{OH1} = −5 mA	V_{DD} − 1.0			V
		I_{OH1} = −100 μA		2.0 V ≤ V_{DD} < 4.0 V	V_{DD} − 0.5			V
	V_{OH2}	引脚 P20 至 P23 的总和 I_{OH2} = −10 mA		4.0 V ≤ AV_{REF} ≤ 5.5 V I_{OH2} = −5 mA	AV_{REF} − 1.0			V
				2.0 V ≤ AV_{REF} < 4.0 V I_{OH2} = −5 mA	AV_{REF} − 0.5			V
输出电压，低	V_{OL}	所有引脚 I_{OL} = 30 mA		4.0 V ≤ V_{DD} ≤ 5.5 V I_{OL} = 10 mA			1.3	V
		2.0 V ≤ V_{DD} < 4.0 V I_{OL} = 400 μA					0.4	V
输入漏电流，高	I_{LIH1}	V_I = V_{DD}		除 X1 以外的引脚			3	μA
输入漏电流，低	I_{LIL1}	V_I = 0 V		除 X1 以外的引脚			−3	μA
输出漏电流，高	I_{LOH}	V_O = V_{DD}		除 X2 以外的引脚			3	μA
输出漏电流，低	I_{LOL}	V_O = 0 V		除 X2 以外的引脚			−3	μA

注　该产品的电压使用范围为 2.2 ～ 5.5 V，因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

备注　除了另外的规定，备用功能引脚的特性与端口引脚的相同。

DC 特性 (T_A = −40 ～ +85°C, V_{DD} = 2.0 ～ 5.5 V[注1], V_{SS} = 0 V) (2/2)

参数	符号	条件			最小	典型	最大.	单位
上拉电阻值	R_{PU}	V_I = 0 V			10	30	100	kΩ
下拉电阻值	R_{PD}	P121，P122，复位状态			10	30	100	kΩ
电源电流[注2]	I_{DD1}[注3]	晶体/陶瓷振荡器, 外部时钟输入振荡器操作模式[注6]	f_X = 10 MHz V_{DD} = 5.0 V ±10%[注4]	A/D 转换器终止操作		6.1	12.2	mA
				A/D 转换器正在操作中[注8]		7.6	15.2	
			f_X = 6 MHz V_{DD} = 5.0 V ±10%[注4]	A/D 转换器终止操作		5.5	11.0	mA
				A/D 转换器正在操作中[注8]			14.0	
			f_X = 5 MHz V_{DD} = 3.0 V ±10%[注5]	A/D 转换器终止操作		3.0	6.0	mA
				A/D 转换器正在操作中[注8]		4.5	9.0	
	I_{DD2}	晶体/陶瓷振荡器, 外部时钟输入 HALT 模式[注6]	f_X = 10 MHz V_{DD} = 5.0 V ±10%[注4]	外设操作终止		1.7	3.8	mA
				外设正在操作中			6.7	
			f_X = 6 MHz V_{DD} = 5.0 V ±10%[注4]	外设操作终止		1.3	3.0	mA
				外设正在操作中			6.0	
			f_X = 5 MHz V_{DD} = 3.0 V ±10%[注5]	外设操作终止		0.48	1	mA
				外设正在操作中			2.1	
	I_{DD3}[注3]	高速 Ring-OSC 操作模式[注7]	f_X = 8 MHz V_{DD} = 5.0 V ±10%[注4]	A/D 转换器终止操作		5.5	11.0	mA
				A/D 转换器正在操作中[注8]		7.0	14.0	
	I_{DD4}	高速 Ring-OSC HALT 模式[注7]	f_X = 8 MHz V_{DD} = 5.0 V ±10%[注4]	外设操作终止		1.4	3.2	mA
				外设正在操作中			5.9	
	I_{DD5}	STOP 模式	V_{DD} = 5.0 V ±10%	低速 Ring-OSC 停止		3.5	35.5	μA
				低速 Ring-OSC 工作		17.5	63.5	
			V_{DD} = 3.0 V ±10%	低速 Ring-OSC 停止		3.5	15.5	μA
				低速 Ring-OSC 工作		11.0	30.5	

注
1. 该产品的电压使用范围为 2.2 ～ 5.5 V，因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。
2. 流经内部电源(V_{DD})的总电流，也包括外设操作电流(但不包括流经端口上拉电阻的电流)。
3. 包括外设操作电流。
4. 当处理器时钟控制寄存器 PCC＝00H。
5. 当处理器时钟控制寄存器 PCC＝02H。
6. 由可选字节选择晶体/陶瓷振荡器,外部时钟输入作为系统时钟源。
7. 由可选字节选择高速 Ring-OSC 作为系统时钟源。
8. 包括流经 AV_{REF} 引脚的电流。

AC 特性

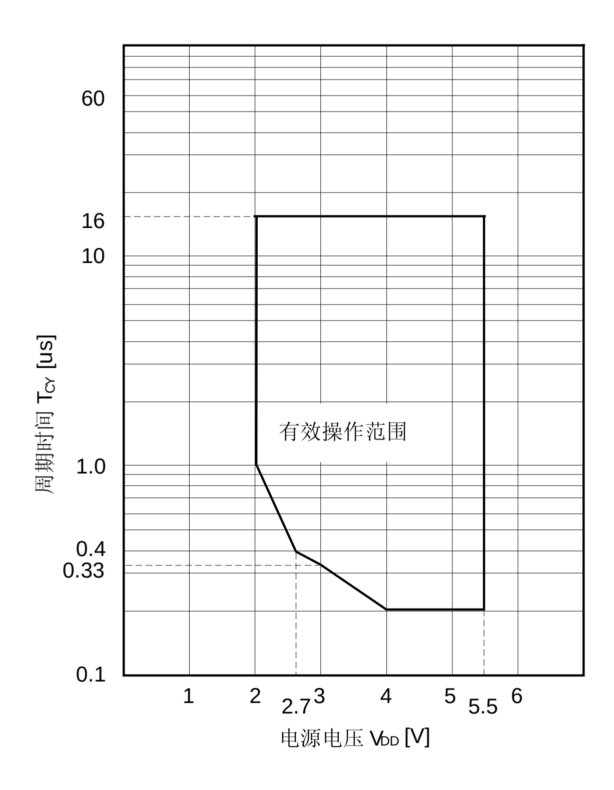
(1) 基本操作 (T_A = −40 ～ +85℃, V_{DD} = 2.0 ～ 5.5 V[注1], V_{SS} = 0 V)

参数	符号	条件		最小	典型	最大	单位
指令周期（指令最短执行时间）	T_{CY}	晶体/陶瓷振荡器,外部时钟输入	4.0 V ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
			3.0 V ≤ V_{DD} < 4.0 V	0.33		16	μs
			2.7 V ≤ V_{DD} < 3.0 V	0.4		16	μs
			2.0 V ≤ V_{DD} < 2.7 V	1		16	μs
		高速 Ring-OSC 时钟	4.0 V ≤ V_{DD} ≤ 5.5 V	0.23		4.22	μs
			2.7 V ≤ V_{DD} < 4.0 V	0.47		4.22	μs
			2.0 V ≤ V_{DD} < 2.7 V	0.95		4.22	μs
TI000 输入高电平宽度, 低电平宽度	t_{TIH}, t_{TIL}	4.0 V ≤ V_{DD} ≤ 5.5 V		2/fsam+ 0.1[注2]			μs
		2.0 V ≤ V_{DD} < 4.0 V		2/fsam+ 0.2[注2]			μs
中断输入高电平宽度, 低电平宽度	t_{INTH}, t_{INTL}			1			μs
$\overline{\text{RESET}}$ 输入低电平宽度	t_{RSL}			2			μs

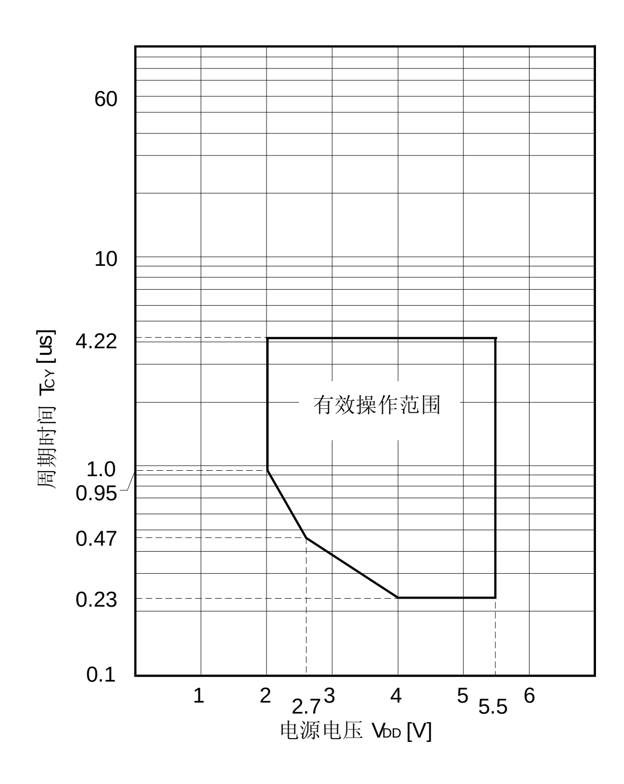
注　1. 该产品的电压使用范围为 2.2 ～ 5.5 V ,因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

　　2. 根据预分频模式寄存器 00 (PRM00)的第 0 和第 1 位(PRM000, PRM001)可以选择 fsam = f_{XP}, f_{XP}/4, 或 f_{XP}/256。注意当选择 TI000 有效沿作为计数时钟时, fsam = f_{XP}。

T_{CY} vs. V_{DD}（晶体/陶瓷振荡器时钟，外部时钟输入）



T_{CY} vs. V_{DD}（高速 Ring-OSC 时钟）



(2) 串行接口 (T_A = −40 ～ +85℃, V_{DD} = 2.0 ～ 5.5 V 注, V_{SS} = 0 V)

UART 模式 (UART6, 专用波特率发生器输出)

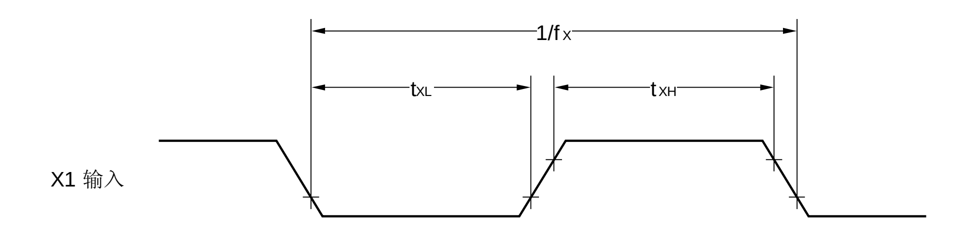
参数	符号	条件	最小	典型	最大.	单位
传输速率					312.5	kbps

注　该产品的电压使用范围为 2.2 ～ 5.5 V ,因为上电清零电路(POC)的检测电压 (V_{POC}) 是 2.1 V ±0.1 V。

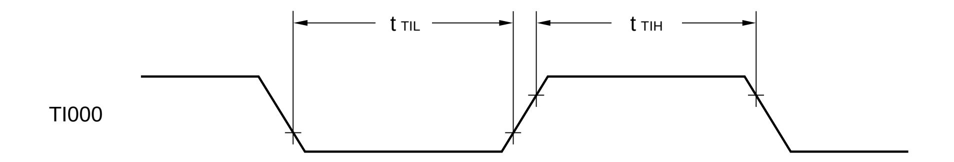
AC 时序测试点 (不包括 X1 输入)



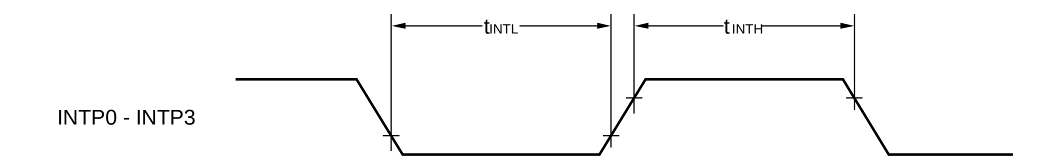
时钟时序



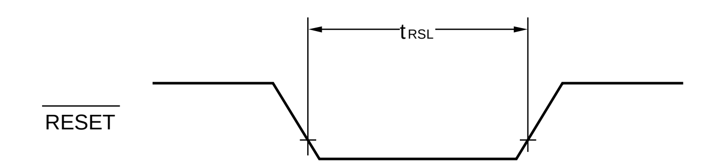
TI000 时序



中断输入时序



$\overline{\text{RESET}}$ 输入时序



A/D 转换器特性 (T_A = −40 ～ +85℃, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小	典型	最大	单位
分辨率			10	10	10	bit
总误差[注s 1, 2]	AINL	4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.6	%FSR
转换时间	t_{CONV}	4.5 V ≤ AV_{REF} ≤ 5.5 V	3.0		100	μs
		4.0 V ≤ AV_{REF} < 4.5 V	4.8		100	μs
		2.85 V ≤ AV_{REF} < 4.0 V	6.0		100	μs
		2.7 V ≤ AV_{REF} < 2.85 V	14.0		100	μs
零度误差[注s 1, 2]	Ezs	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
满度误差[注s 1, 2]	Efs	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
积分非线性误差[注 1]	ILE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
微分非线性误差[注 1]	DLE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
模拟输入电压	V_{AIN}		V_{SS}		AV_{REF}	V

注 1. 不包括量化误差 (±1/2 LSB)。

2. 该值表示满度的百分比 (%FSR) 。

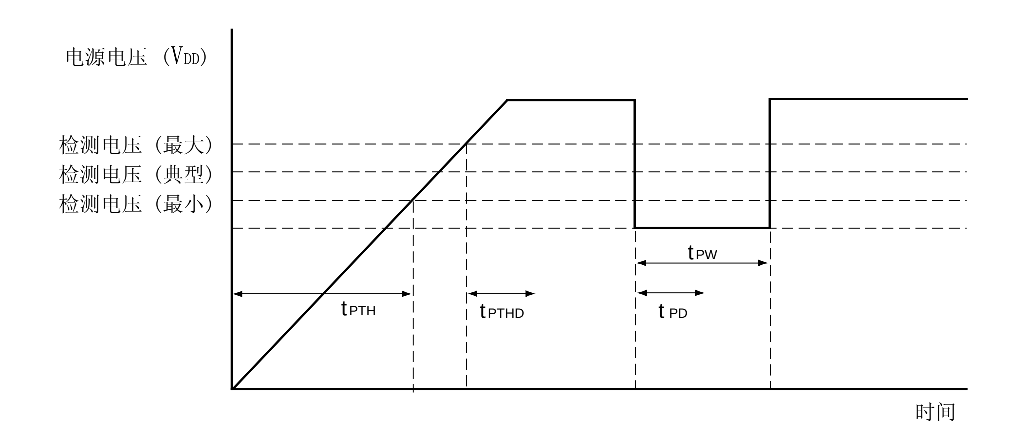
POC 电路特性 (T_A = −40 ～ +85°C)

参数	符号	条件	最小	典型	最大	单位
检测电压	V_{POC}		2.0	2.1	2.2	V
电源启动时间	t_{PTH}	V_{DD}: 0 V → 2.1 V	1.5			μs
响应延迟时间 1 [注1]	t_{PTHD}	上电时，并达到检测电压(最大值) 后			3.0	ms
响应延迟时间 2 [注2]	t_{PD}	下电时			1.0	ms
最短脉冲宽度	t_{PW}		0.2			ms

注　1.　从电压检测到复位释放所需要的时间。

　　　2.　从电压检测到内部复位信号产生所需要的时间。

POC 电路时序



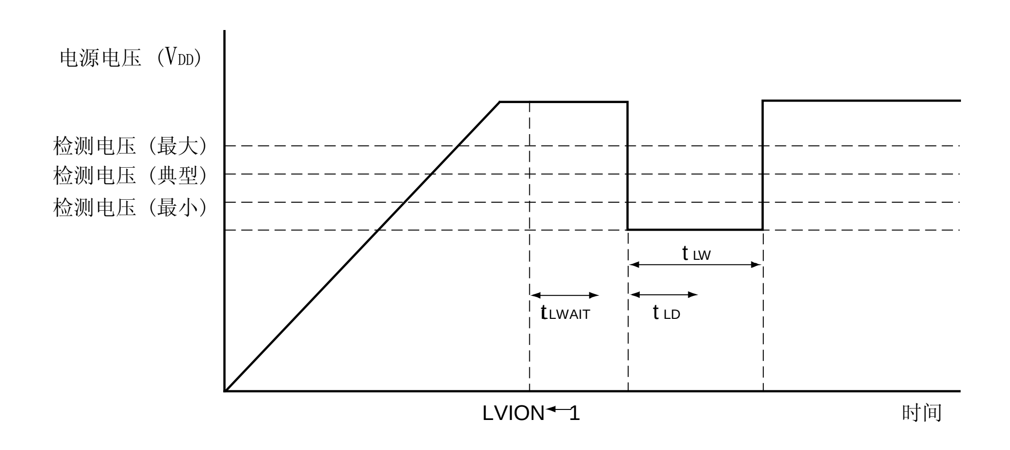
LVI 电路特性 (T_A = –40 ～ +85°C)

参数	符号	条件	最小	典型	最大	单位
检测电压	V_{LVI0}		4.1	4.3	4.5	V
	V_{LVI1}		3.9	4.1	4.3	V
	V_{LVI2}		3.7	3.9	4.1	V
	V_{LVI3}		3.5	3.7	3.9	V
	V_{LVI4}		3.3	3.5	3.7	V
	V_{LVI5}		3.15	3.3	3.45	V
	V_{LVI6}		2.95	3.1	3.25	V
	V_{LVI7}		2.7	2.85	3.0	V
	V_{LVI8}		2.5	2.6	2.7	V
	V_{LVI9}		2.25	2.35	2.45	V
响应时间注 1	t_{LD}			0.2	2.0	ms
最小脉冲宽度	t_{LW}		0.2			ms
操作稳定等待时间注 2	t_{LWAIT}			0.1	0.2	ms

注 1. 从电压检测到中断输出或内部复位产生所需的时间。
2. 从设置 LVION ＝ 1 到操作稳定所需的时间。

备注 1. V_{LVI0} ＞ V_{LVI1} ＞ V_{LVI2} ＞ V_{LVI3} ＞ V_{LVI4} ＞ V_{LVI5} ＞ V_{LVI6} ＞ V_{LVI7} ＞ V_{LVI8} ＞ V_{LVI9}
2. V_{POC} ＜ V_{LVIm} (m = 0 ～ 9)

LVI 电路时序



数据存储器 STOP 模式低电源电压下数据保持特性 (T_A = –40～ +85°C)

参数	符号	条件	最小	典型	最大	单位
数据保持电源电压	V_{DDDR}		2.0		5.5	V
释放信号设置时间	t_{SREL}		0			μs

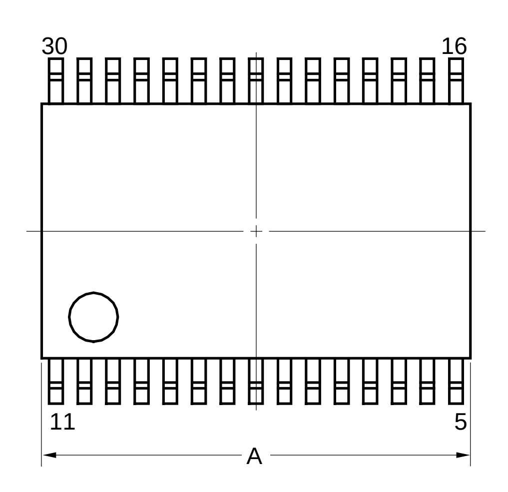
Flash 存储器编程特性 (T_A = －40 ～ +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

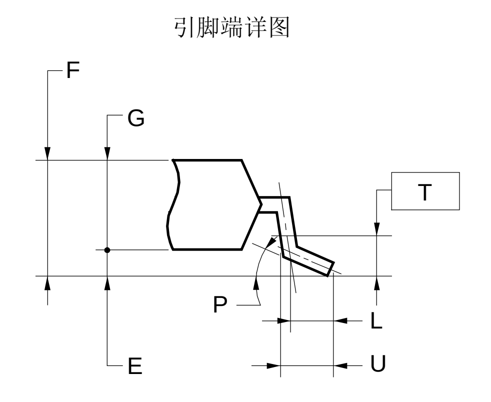
参数	符号	条件		最小	典型	最大	单位
电源电压	I_{DD}	V_{DD} = 5.5 V				7.0	mA
擦除次数注 (每 1 个块)	N_{ERASE}	T_A = −10 ～ +85℃				1000	次
		T_A = −40 ～ +85℃				T.B.D.	次
芯片擦除时间	T_{CERASE}	T_A = −10 ～ +85℃, N_{ERASE} ≤ 100	4.5 V ≤ V_{DD} ≤ 5.5 V			0.90	s
			3.5 V ≤ V_{DD} < 4.5 V			1.00	s
			2.7 V ≤ V_{DD} < 3.5 V			1.20	s
		T_A = −10 ～ +85℃, N_{ERASE} ≤ 1000	4.5 V ≤ V_{DD} ≤ 5.5 V			3.52	s
			3.5 V ≤ V_{DD} < 4.5 V			3.92	s
			2.7 V ≤ V_{DD} < 3.5 V			4.69	s
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 100	4.5 V ≤ V_{DD} ≤ 5.5 V			T.B.D.	s
			3.5 V ≤ V_{DD} < 4.5 V			T.B.D.	s
			2.7 V ≤ V_{DD} < 3.5 V			T.B.D.	s
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 1000	4.5 V ≤ V_{DD} ≤ 5.5 V			T.B.D.	s
			3.5 V ≤ V_{DD} < 4.5 V			T.B.D.	s
			2.7 V ≤ V_{DD} < 3.5 V			T.B.D.	s
Block 擦除时间	T_{BERASE}	T_A = −10 ～ +85℃, N_{ERASE} ≤ 100	4.5 V ≤ V_{DD} ≤ 5.5 V			0.48	s
			3.5 V ≤ V_{DD} < 4.5 V			0.53	s
			2.7 V ≤ V_{DD} < 3.5 V			0.63	s
		T_A = −10 ～ +85℃, N_{ERASE} ≤ 1000	4.5 V ≤ V_{DD} ≤ 5.5 V			1.86	s
			3.5 V ≤ V_{DD} < 4.5 V			2.07	s
			2.7 V ≤ V_{DD} < 3.5 V			2.48	s
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 100	4.5 V ≤ V_{DD} ≤ 5.5 V			T.B.D.	s
			3.5 V ≤ V_{DD} < 4.5 V			T.B.D.	s
			2.7 V ≤ V_{DD} < 3.5 V			T.B.D.	s
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 1000	4.5 V ≤ V_{DD} ≤ 5.5 V			T.B.D.	s
			3.5 V ≤ V_{DD} < 4.5 V			T.B.D.	s
			2.7 V ≤ V_{DD} < 3.5 V			T.B.D.	s
字节写时间	T_{WRITE}	T_A = −10 ～ +85℃, N_{ERASE} ≤ 1000				150	μs
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 1000				T.B.D.	μs
内部校验时间	T_{VERIFY}	每块				6.8	ms
		每字节				27	μs
空白检测	T_{BLKCHK}	每块				480	μs
保存年限		T_A = −10 ～ +85℃, N_{ERASE} ≤ 1000		10			年
		T_A = −40 ～ +85℃, N_{ERASE} ≤ 1000		T.B.D.			年

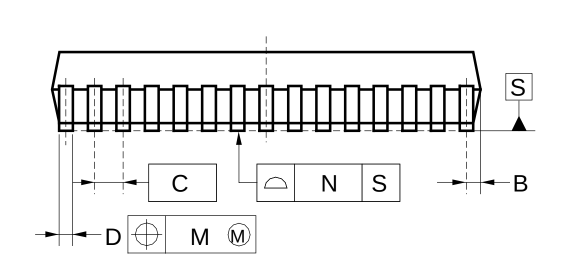
注　根据擦除次数不同(N_{ERASE})，擦除时间也不同。参考芯片擦除时间和块擦除时间参数。

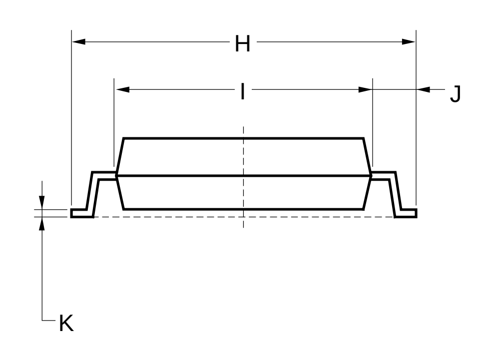
备注　在产品出货后初次写入时，“擦除 → 写入” 和 “只写” 都被认为是一次重写。

30引脚塑封 SSOP (7.62 mm (300))









注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置(T.P.)的0.13mm 内

项目	毫米
A	9.85 ±0.15
B	0.45 最大
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1 ±0.05
F	1.3 ±0.1
G	1.2
H	8.1 ±0.2
I	6.1 ±0.2
J	1.0 ±0.2
K	0.17 ±0.03
L	0.5
M	0.13
N	0.10
P	$3°^{+5°}_{-3°}$
T	0.25
U	0.6 ±0.15

S30MC-65-5A4-2

附录 A　开发工具

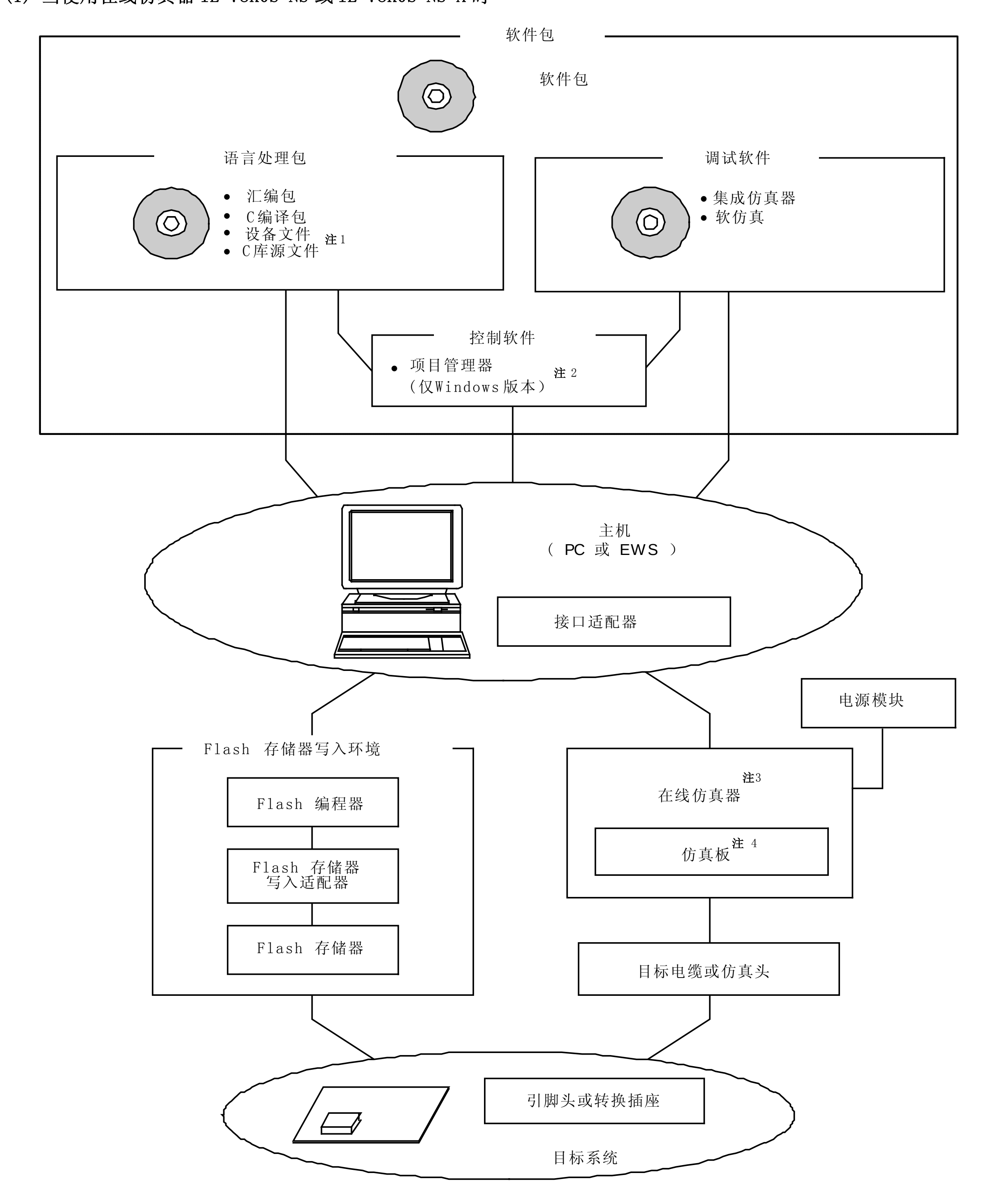
下面的开发工具在使用 78K0S/KB1+的系统开发中使用。开发工具如图 A-1 所示。

- PC98-NX 系列兼容机
  除非特别说明,IBM PC/AT™支持的产品和 PC98-NX 系列兼容。当使用 PC98-NX 系列时可参考 IBM PC/AT 及其兼容机的使用说明。

- Windows™
  除非特别说明，“Windows” 是指以下操作系统：
  - Windows 98
  - Windows NT™ Ver. 4.0
  - Windows 2000
  - Windows XP

图 A-1. 开发工具 (1/2)

(1) 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时



注 1. 软件包中不包括 C 库源文件。

2. 汇编包中包括项目管理器 PM+。

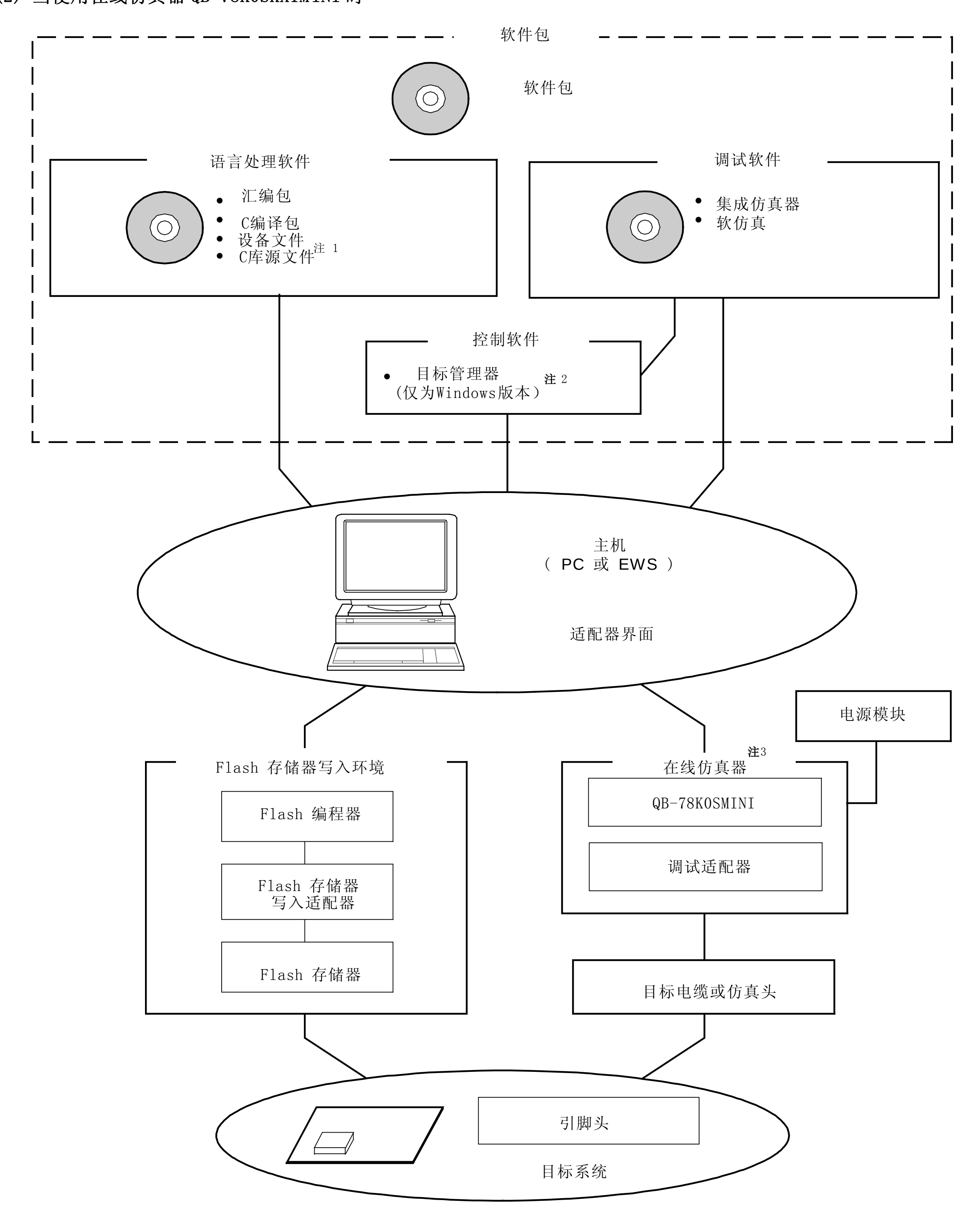
PM+ 只能在 windows 环境下使用。

3. 除了在线仿真器 IE-78K0S-NS 和 IE-78K0S-NS-A 外的其他产品都是单独销售的。

4. 在线仿真器 IE-789234-NS-EM1 提供目标电缆。

图 A-1. 开发工具 (2/2)

(2) 当使用在线仿真器 QB-78K0SKX1MINI 时



注 1. 软件包中不包括 C 库源文件。

2. 汇编包中包括项目管理器 PM+。
PM+ 应用于 windows 环境。

3. 在线仿真器 QB-78K0SKX1MINI 提供集成调试环境 ID78K0S-QB，flash 编程器 PG-FPL2，电源模块和目标电缆，其他产品都单独销售。

A.1 软件包

SP78K0S 软件包	此软件包用于 78K/0S 系列的开发。 包含如下工具： RA78K0S，CC78K0S，ID78K0S-NS 等。
	产品编号：μS××××SP78K0S

备注 产品编号中的×××× 随使用的操作系统而定。


μS××××SP78K0S

××××	主机	操作系统	支持媒介
AB17	PC-9800 系列,IBM PC/AT 及其兼容机	日文 Windows	CD-ROM
BB17		英文 Windows	

A.2 语言处理软件

RA78K0S 汇编包	该程序将编写的助记符程序转换为可在微控制器中运行的目标代码。 可自动产生符号表和优化分支指令。 与设备文件(DF789234) (单独销售)联合使用。 **<PC 环境使用的注意事项>** 该汇编包是基于 DOS 运行的，但使用 PM+ (包含在汇编包中)可在 windows 中使用。
	产品编号： μS××××RA78K0S
CC78K0S C 语言编译	该程序将编写的 C 语言程序转换为可在微控制器中运行的目标代码。 与汇编包(RA78K0S) 和设备文件 (DF789234) (二者均独立销售)联合使用。 **< PC 环境使用的注意事项 >** C 编译包是基于 DOS 运行的，但使用 PM+ (包含在汇编包中)可在 windows 中使用。
	产品编号： μS××××CC78K0S
DF789234 注 1 设备文件	该文件包含该器件特有的信息。 与其他工具联合使用（RA78K0S，CC78K0S， ID78K0S-NS，ID78K0S-QB 或 SM+ for 78K0S）。
	产品编号： μS××××DF789234
CC78K0S-L 注 2 C 库源文件	函数库的源文件，可根据用户的特殊需求去改变 C 编译库中的目标库。 由于是源文件，它的使用不随操作系统改变。
	产品编号： μS××××CC78K0S-L

注 1. DF789234 是一个通用文件，可被 RA78K0S,CC78K0S,ID78K0S-NS,ID78K0S-QB 和 SM+ for 78K0S 使用。
2. SP78K0S 软件包不包括 CC78K0S-L。

备注 产品编号中的×××× 随使用的主机和操作系统而定。

μS××××RA78K0S
μS××××CC78K0S
μS××××CC78K0S-L

××××	主机	操作系统	支持媒介
AB17	PC-9800 系列, IBM PC/AT 及其兼容机	日文 Windows	CD-ROM
BB17		英文 Windows	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

μS××××DF789234

××××	主机	操作系统	支持媒介
AB13	PC-9800 系列, IBM PC/AT 及其兼容机	日文 Windows	3.5” 2HD 软盘
BB13		英文 Windows	

A.3 控制软件

PM+ PM	这个控制软件可以使用户在 windows 下高效的开发。启动编辑器, 构造程序,启动调试器, 都能够在 PM+下执行。 **<注意事项>** PM+ 包含在汇编包 (RA78K0S)中。 只能在 windows 环境运行。

A.4 Flash 存储器写入工具

FlashPro4 (FL-PR4, PG-FP4) Flash 存储器编程器	只适用于 flash 型程序存储器的微控制器。
PG-FPL2 Flash 存储器编程器	只适用于 flash 型程序存储器的微控制器。 在线仿真器 QB-78K0SKX1MINI 提供。
FA-30MC-5A4-A Flash 存储器编程器适配器	Flash 存储器写入适配器。用于和 Flash 编程器的连接。 使用 30 引脚塑封 SSOP (MC-5A4 型)。

备注 FL-PR4 和 FA-30MC-5A4-A 是 Naito Densei Machida Mfg. Co., Ltd. 的产品。
如需了解更多信息,联系:Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)

A.5 调试工具（硬件）

A.5.1 当使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A

产品	说明
IE-78K0S-NS 在线仿真器	用于在线调试 78K/0S 系列的硬件和软件应用。支持集成调试器(ID78K0S-NS)。与 AC 适配器，仿真插头，和用于连接主机的接口配器联合使用。
IE-78K0S-NS-A 在线仿真器	覆盖 IE-78K0S-NS 的所有功能， 并且提高了调试功能，如跟踪、定时器功能。
IE-70000-MC-PS-B AC 适配器	支持从 100 到 240 VAC 的电源。
IE-70000-CD-IF-A PC 接口卡	当使用笔记本电脑(支持 PCMCIA 接口)做主机时需要的接口卡和电缆。
IE-70000-PC-IF-C 接口适配器	IBM PC/AT 及其兼容机(支持 ISA 总线)做主机时 需要的接口适配器。
IE-70000-PCI-IF-A 接口适配器	有 PCI 总线的个人计算机做主机时需要的接口适配器。
IE-789234-NS-EM1 仿真板	仿真板用于仿真器件的外围硬件。 联合在线仿真器使用，提供一根目标电缆。
NP-30MC 仿真插头	用于连接在线仿真器和目标板，这个插头使用 30 引脚塑封 SSOP (MC-5A4 型)。
NSPACK30BK YSPACK30BK 转换连接器	转换连接器将 NP-30MC 和安装 30 引脚塑封 SSOP（MC-5A4 型）的目标板进行连接。 • NSPACK30BK: 连接到目标板的连接器 • YSPACK30BK: 连接到仿真器的连接器
QB-80-EP-01T 仿真插头	用于连接在线仿真器和目标系统，这个插头使用 30 引脚塑封 SSOP (MC-5A4 型)。
QB-30MC-EA-01T	这个转换适配器用于进行从在线仿真器到目标连接器的引脚转换。
QB-30MC-YQ-01T	这个 YQ 连接器用于连接目标连接器和转换适配器。
QB-30MC-NQ-01T	这个目标连接器用于装配在目标系统上。
目标系统引脚接头的规格	0.635 mm × 0.635 mm (高: 6 mm)

备注 1. NP-30MC 是 Naito Densei Machida Mfg. Co., Ltd 的产品。
了解更多信息，联系：Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191)
2. NSPACK30BK 和 YSPACK30BK 是 TOKYO ELETECH CORPORATION 的产品。
了解更多信息，联系：Daimaru Kogyo Co., Ltd.
Tokyo Electronics Department (TEL: +81-3-3820-7112)
Osaka Electronics Department (TEL: +81-6-6244-6672)

A.5.2 当使用在线仿真器 QB-78K0SKX1MINI 时

产品	说明
QB-78K0SKX1MINI 在线仿真器	在线仿真器用于调试使用 78K0S/Kx1+系列的硬件和软件。支持集成调试器（ID78K0S-QB）。与 AC 适配器，目标电缆，连接主机的 USB 接口电缆联合使用。
目标系统引脚接头的规格	0.635 mm × 0.635 mm (高: 6 mm)

A.6 调试工具（软件）

产品	说明
ID78K0S-NS（支持在线仿真器 IE-78K0S-NS/IE-78K0S-NS-A）集成调试器	用于 78K/0S 系列的在线仿真。ID78K0S-NS 是基于 windows 的软件。 该调试器增强了 C 语言的调试功能。通过使用集成窗口功能，可以将源程序， 反汇编显示， 和带跟踪结果的存储器显示联系在一起， 可以显示对应于源程序的跟踪结果。与设备文件联合使用（DF789234）（独立销售）。

	产品编号： μS××××ID78K0S-NS
ID78K0S-QB（支持在线仿真器 QB-78K0SKX1MINI）集成调试器	用于 78K0S/Kx1+ 系列的在线仿真。ID78K0S-QB 是基于 windows 的软件。 提供 C 语言调试功能，可以显示源程序，反汇编显示，存储器显示。和设备文件联合使用（DF789234）（独立销售）。 在线仿真器 QB-78K0SKX1MINI 中提供。
	产品编号： μS××××ID78K0S-QB（不单独销售）
SM+ for 78K0S 注 1 软件仿真器	用于 78K/0S 系列的软件仿真。SM+ for 78K0S 是基于 windows 的软件。 在主机上模拟目标系统的操作，可执行 C 语言级和汇编语言级的调试。 使用 SM+ for 78K0S,可以在不需要外围硬件的情况下验证程序逻辑和功能，因此可以增强开发效率改进软件质量。 和设备文件配合使用（DF789234）（独立销售）。
	产品编号： μS××××SM789234-B
DF789234 注 2 设备文件	该文件包含该器件特有的信息。 与其他工具联合使用（RA78K0S，CC78K0S，ID78K0S-NS，ID78K0S-QB 和 SM+ for 78K0S）（全部独立销售）。
	产品编号： μS××××DF789234

注 1. 开发中

2. DF789234 是一个通用文件，可被 RA78K0S，CC78K0S，ID78K0S-NS，ID78K0S-QB 和 SM+ for 78K0S 使用。

备注 产品编号中的×××× 随使用的操作系统和支持媒介而定。

μS××××ID78K0S-NS
μS××××ID78K0S-QB
μS××××SM789234-B

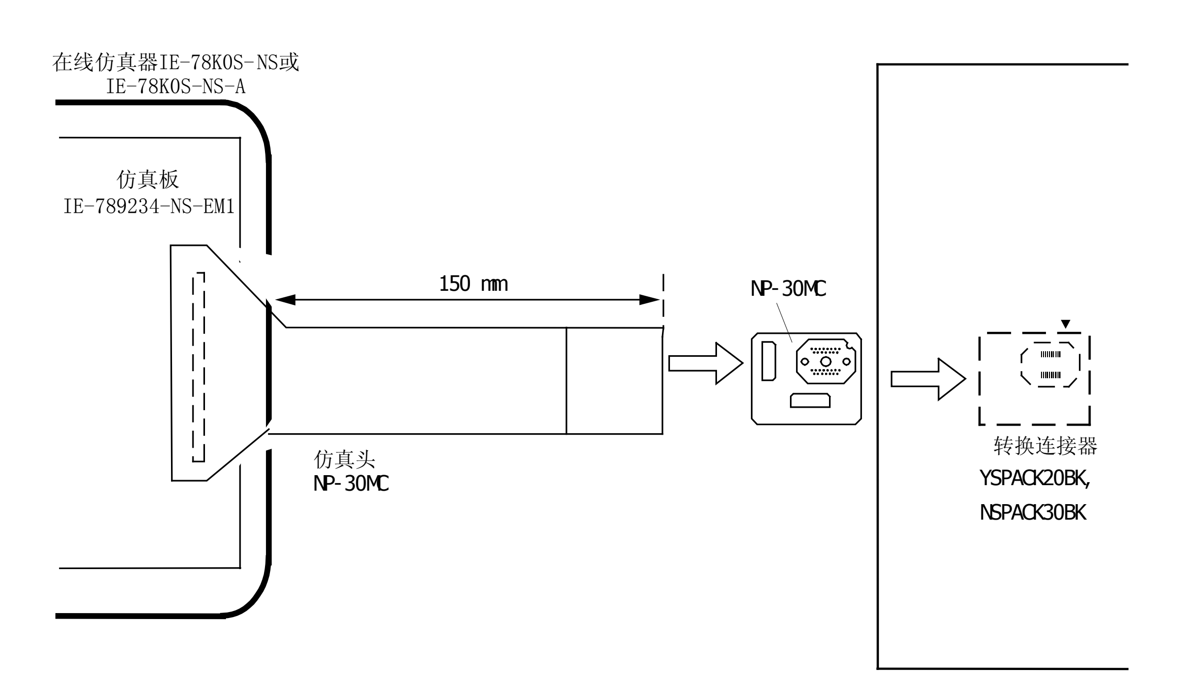
××××	主机	操作系统	支持媒介
BB13	PC-9800 系列,IBM PC/AT 及其兼容机	英文 Windows	3.5” 2HD 软盘
AB17		日文 Windows	CD-ROM
BB17		英文 Windows	

μS××××DF789234

××××	主机	操作系统	支持媒介
AB13	PC-9800 系列,IBM PC/AT 及其兼容机	日文 Windows	3.5” 2HD 软盘
BB13		英文 Windows	

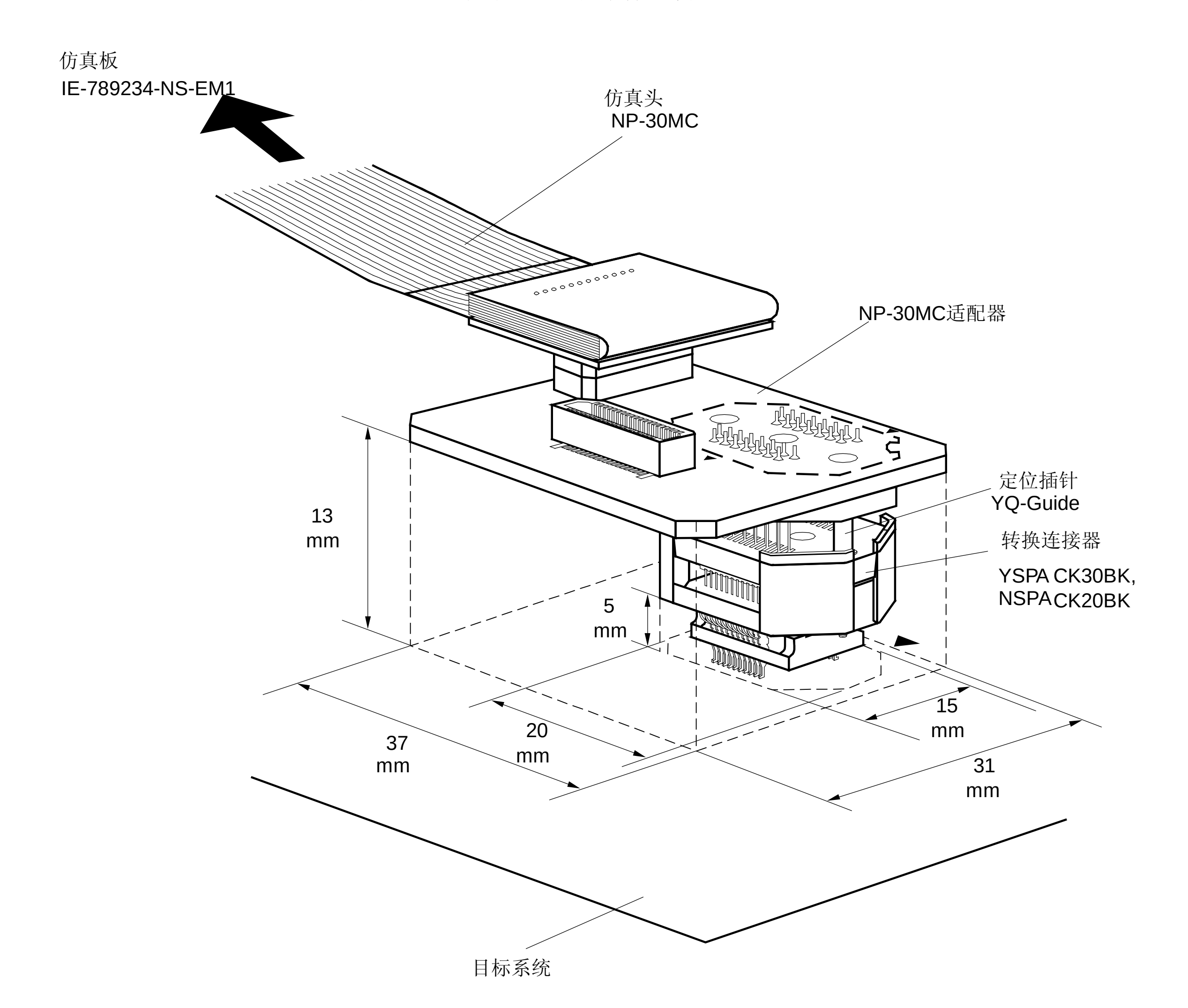
下图显示了使用在线仿真器 IE-78K0S-NS 或 IE-78K0S-NS-A 时仿真探头到转换连接器和转换插座之间的连接条件。设计系统时必须满足一些条件，如安装在目标系统上的部件形状，如下所示。

图 B-1. 在线仿真器和转换连接器之间的距离（使用 NP-30MC 时）



备注 **1.** NP-30MC 是 Naito Densei Machida Mfg. Co., Ltd 的产品。
2. YSPACK30BK 和 NSPACK30BK 是 TOKYO ELETECH CORPORATION 的产品。

图 **B-2.** 目标系统的连接条件（使用 **NP-30MC** 时）



备注 **1.** NP-30MC 是 Naito Densei Machida Mfg. Co., Ltd 的产品。

2. YSPACK30BK，NSPACK30BK 和 YQGUIDE 是 TOKYO ELETECH CORPO-RATION 的产品。

图 **B-3.** 在线仿真器和转换连接器之间的距离（使用 **QB-80-EP-01T** 时）

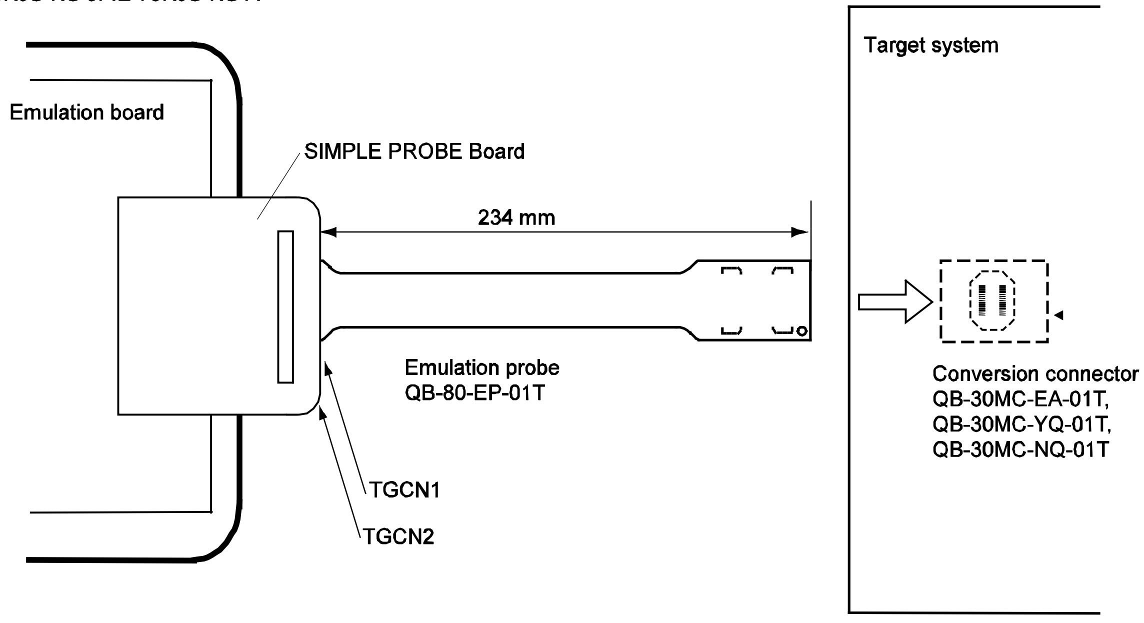
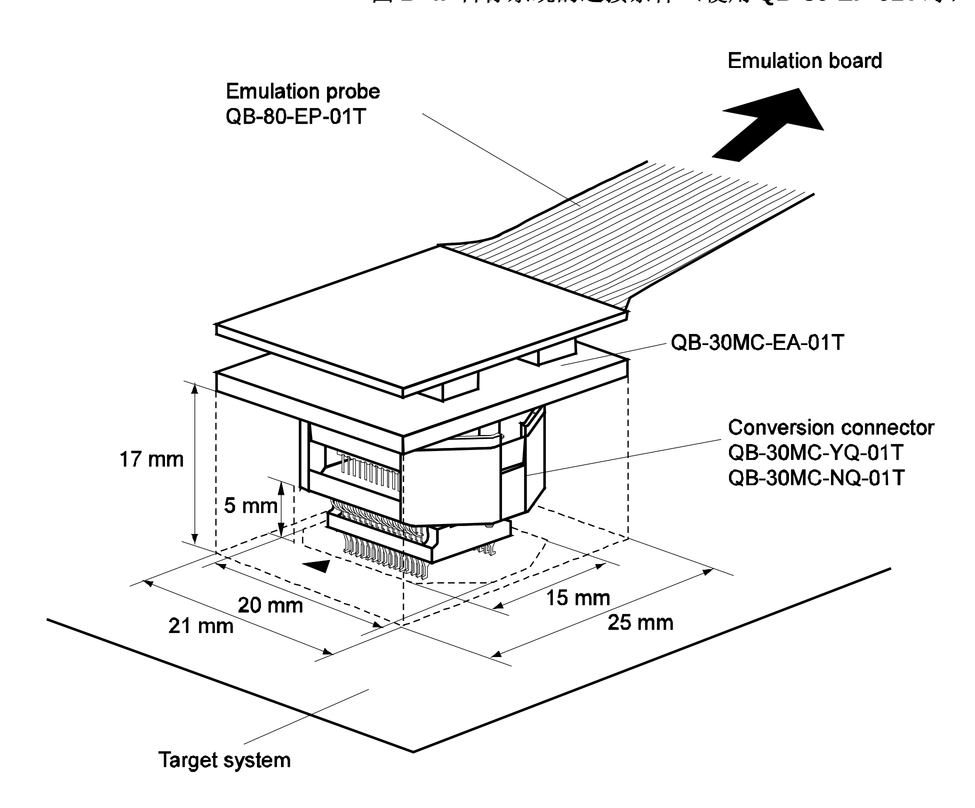


图 **B-4.** 目标系统的连接条件（使用 **QB-80-EP-01T** 时）



附录 C　寄存器索引

C.1　寄存器索引(按寄存器名称排序)

C.2 寄存器索引(按寄存器符号排序)

附录 D 注意事项列表

本附录列表本文档中的注意事项。
表中“类别（硬件/软件）”定义如下。

硬件： 微控制器内部/外部硬件的注意事项
软件： 寄存器设置或编程的注意事项

(1/18)

章节	类别	功能	功能详解	注意事项	页号
第一章	硬件	引脚配置	AVss 引脚	将 AVss 引脚和 Vss 引脚连接在一起。	p.17 □
第二章	硬件	引脚功能	P121/X1 引脚和 P122/X2 引脚	P121/X1 和 P122/X2 引脚复位期间为下拉状态。	p. 21, 22, 24, 25 □
第三章	硬	存储器空间	向量表地址	向量表地址 0014H 不对应任何中断源。	p.30 □
	软		SP：堆栈指针	由于复位输入使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。	p.34 □
第四章	硬件	端口功能	P121/X1 引脚和 P122/X2 引脚	P121/X1 和 P122/X2 引脚复位期间为下拉状态。	p.50 □
			P34 引脚	因为引脚 P34 可以作为 RESET 引脚使用，如果当它作为输入端口引脚使用的时候，从外部输入一个复位信号也不能实现复位功能。这个端口的功能通过可选字节来选择，具体细节清参照第十八章 可选字节。 果复位状态被 POC 电路释放后，再次参照可选字节之前，一个低电平被输入到 RESET 引脚， 78K0S/KB1+被复位并保持在复位状态直到一个高电平输入到 RESET 引脚。	p.56 □
			P30，P31 和 P43 引脚	因为 P30, P31 和 P43 引脚也被用作外部中断引脚，这些引脚的任何一个被设置成输出模式并它的输出电平被改变，则相应的中断请求标志被置位。因此使用这个端口为输出模式时，预先设置相应的中断屏蔽标志为 1。	p.64 □
			-	虽然 1 位存储器操作指令只对 1 位进行操作，但实际是以 8 位形式访问端口。因此，如果一个端口既可输入，又可输出，则对于端口中定义为输入模式的引脚的输出锁存器中的内容是不确定的。	p.68 □
第五章	软件	系统时钟	PCC： 处理器时钟控制寄存器	第 7 位到第 2 位和第 0 位必须设置为 0。	p.72 □

(2/18)

章节	类别	功能	功能详解	注意事项	页号
第五章	软件	主时钟	OSTS：振荡稳定时间选择寄存器	设置和释放 STOP 模式时按照下面的方法设置振荡稳定时间：OSTS 设置的振荡稳定时间大于等于预期的振荡器振荡稳定时间。	p.74 □
				无论 STOP 模式是被复位信号打破还是被中断打破，STOP 模式释放后的等待时间不包括从释放 STOP 模式到晶振开始震荡的时间（下图中的 a）	p.74 □
				上电或者复位后的振荡稳定时间可以通过可选字节来设置，详情参见 第十八章 可选字节。	p.74 □
	硬件	晶体/陶瓷振荡器	–	当使用晶体/陶瓷振荡器时，布线应该按照图 5-6 中虚线内的方式连接，这样可以减小分布电容对振荡器的影响。 • 保持连线尽量的短。 • 不要使其它的信号线与振荡器导线互相交叉，不要在振荡器旁边通过大电流的信号线。 • 保持振荡器电容的地和 Vss 同电位，不要让电容使用的地通过大电流。 • 不要从振荡器上取信号。	p.75 □
第六章	硬件	16 位定时器/事件计数器 00	TM00：16 位定时计数器 00	TM00 被读取时, CR010 无法捕捉其值。	p.87 □
				当读 TM00 时，计数时钟不会停止。	p.87 □
	软件		CR000：16 位定时器捕捉/比较寄存器 000	TM00 和 CR000 相等时进入的清零&启动模式中，CR000 置非 0000H 值。当此寄存器作为外部事件计数器使用时，不能实现仅对一个脉冲的计数。然而,在自由运行模式和由 TI000 引脚有效沿产生的清零&启动模式中, 如果 CR000 置 0000H，TM00 溢出后从 0000H 变为 0001H 时，产生一个中断请求 (INTTM000)。	p.88 □
				如果 CR000 的新值小于 16 位定时器计数器 0 (TM00)的值，TM00 继续计数，溢出，然后从 0 开始重新计数。因此，如果 CR000 的新值小于原来的值，在 CR000 的值改变后，定时器必须复位重新启动。	p.88 □
				当 16 位定时器/事件计数器 00 停止工作后，CR000 的值是不确定的。	p.88 □
	硬件			当 CR000 设为比较模式，即使有捕捉触发信号输入，也不响应捕捉操作。	p.88 □
				当 P31 引脚作为 TI010 有效沿输入端口使用时，P31 引脚则不能作为定时器输出(TO00)使用。　而且，当 P31 用作 TO00 时，也不能作为 TI010 有效沿输入引脚使用。	p.88 □

(3/18)

章节	类别	功能	功能详解	注意事项	页号
第六章	硬件	16 位定时器/事件计数器 00	CR000：16 位定时器捕捉/比较寄存器 000	当 CR000 作为捕捉寄存器使用，如果同时发生寄存器的读操作和输入捕捉触发操作，则捕捉触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕捉触发输入同时发生，捕捉触发为不确定状态。	p.88 □
	软件			改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00（17）在定时器操作中改变比较寄存器的注意事项。	p.88 □
			CR010：16 位定时器捕捉/比较寄存器 010	在自由运行模式和由 TI000 引脚有效沿产生的清零&启动模式中，如果 CR010 置 0000H，TM00 溢出后从 0000H 变为 0001H 时，产生一个中断请求（INTTM010）。	p.89 □
				如果 CR010 新值小于 16 位定时器计数器 0（TM00）的值，TM00 继续计数，溢出，然后从 0 开始重新计数。因此，如果 CR010 新值小于原来的值，则必须在改变赋值后重新启动定时器。	p.89 □
				当 16 位定时器/事件计数器 00 停止工作后，CR010 的值是不确定的。	p.89 □
	硬件			当 CR010 设为比较模式，即使有捕捉触发信号输入，也不响应捕捉操作。	p.89 □
				当 CR010 作为捕捉寄存器使用，如果同时发生寄存器的读操作和输入捕捉触发操作，则捕捉触发输入有优先权，而读出的数据为不确定值。同样，如果定时器停止计数与捕捉触发输入同时发生，捕捉触发为不确定状态。	p.89 □
	软件			改变 CR010 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00（17）在定时器操作中改变比较寄存器的注意事项。	p.89 □
			TMC0016 位定时器模式控制寄存器 00	改变 CR010 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00（17）在定时器操作中改变比较寄存器的注意事项。	p.90 □
				除了 OVF00 标志位，其他位写入数据时，定时器必须停止操作。	p.91 □
	硬件			如果定时器停止，即使 TI000/TI010 引脚有信号输入，定时器计数和中断也不响应。	p.91 □
	软件			除非当 TI000 引脚有效沿用于计数时钟，在设置定时器 STOP 模式或系统时钟停止之前，必须停止定时器工作，否则，当系统时钟启动时定时器可能产生错误。	p.91 □
				当定时器停止工作后才能通过预分频模式寄存器 00(PRM00)的第 4 位和第 5 位设置 TI000 引脚有效沿。	p.91 □
				在 TM00 和 CR00 相等时进入的清零&启动模式中，当定时器工作在自由工作模式时，当 CR000 的值为 FFFFH，TM00 的值从 FFFFH 变为 0000H 后，OVF00 标志位置 1。	p.91 □

(4/18)

章节	类别	功能	功能详解	注意事项	页号
第六章	软件	16 位定时器/事件计数器 00	TMC00：16 位定时器模式控制寄存器 00	即便在 TM00 溢出后而下一个计数时钟计数前（TM00 变为 0001H 前）做清除 OVF00 标志操作， OVF00 标志还会被重新置 1， 清除操作无效。	p.91 □
				捕捉操作在计数时钟的下降沿执行，中断请求（INTTM0n0）在计数时钟的上升沿产生。	p.91 □
			CRC00：捕捉/比较控制寄存器 00	设置 CRC00 前，定时器必须停止操作。	p.92 □
				当通过 16 位定时器模式控制寄存器 TMC00 选择为 TM00 和 CR000 相等进入的清零&启动模式时， CR000 就不能被指定为捕捉寄存器。	p.92 □
	硬件			为了确保捕捉操作的稳定，被捕捉的信号宽度要大于 2 个计数时钟(由预分频器模式寄存器 00 (PRM00)选择)的长度才能触发捕捉 (参见图 6-17)。	p.92 □
	软件		TOC00：16 位 定时器输出控制寄存器 00	除了 OSPT00 以外，对其它寄存器设置之前，定时器应停止工作。	p.93 □
				读 LVS00 和 LVR00 时，读出的数据为 0。	p.93 □
				数据被设置后， OSPT00 自动清零，读出的内容是 0 。	p.93 □
				除了单脉冲输出模式以外，不要将 OSPT00 设置为 1。	p.93 □
	硬件			当连续设置 OSPT00 位时，需要两个或更多计数时钟周期(由预分频器模式寄存器 00 (PRM00)选择)的写入间隔。	p.93 □
	软件			当 TOE00 为 0 时， 设置 TOE00， LVS00， 和 LVR00 要用 8 位存储器操作指令同时设置。 当 TOE00 为 1 时， LVS00 ，LVR00 能够用 1 位存储器操作指令设置。	p.93 □
			PRM00：预分频模式寄存器 00	停止定时器工作后，向 PRM00 写入数据。	p.95 □
				如果 TI000 引脚的有效沿被用于计数时钟， 就不能设置清零启动模式和 TI000 引脚有效沿的捕捉。	p.95 □
	硬件			在下列情况下，注意 TI0n0 引脚有效沿被检测的警告 〈1〉 系统复位后，如果一个高电平输入 TI0n0 引脚,16 位定时器计数器 00 (TM00)允许操作。 →如果上升沿或双沿被指定为 TI0n0 引脚的有效沿，TM00 操作允许后，马上会检测到一个上升沿。 〈2〉 当 TI0n0 为高电平，TM00 停止工作时，TI0n0 引脚如输入一个低电平则 TM00 开始工作。 →如果 TI0n0 的下降沿或上升沿和下降沿作为有效沿，那么 TM00 操作使能后，下降沿会立即检测到。 〈3〉 当 TI0n0 为低电平，TM00 停止工作时，TI0n0 引脚如输入一个高电平则 TM00 开始工作。 →如果 TI0n0 的上升沿或上升沿和下降沿作为有效沿，那么 TM00 操作使能后，上升沿会立即检测到。	p.95 □

(5/18)

章节	类别	功能	功能详解	注意事项	页号
第六章	硬件	16 位定时器/事件计数器 00	PRM00：预分频模式寄存器 00	当 TI000 的有效沿用做计数时钟，和当 TI000 用于捕捉触发器时，用于消除噪声的采样时钟是不同的，对于前者，由 f_{XP} 做计数时钟，对于后者，通过 PRM00 选择的计数时钟。直到采样到有效沿并且连续两次检测到有效电平，捕捉操作才会完成，因此可消除小脉冲宽度的噪声。	p.95 □
				当 P31 端口作为有效沿的输入引脚(TI010)使用时，它不能同时作为定时器输出引脚(TO00)使用。当 P31 端口作为定时器输出引脚(TO00)使用时，它不能同时作为有效沿的输入引脚(TI010)使用	p.95 □
	软件		间隔定时	当 TM00 工作时，改变 CR000 寄存器的设置可能导致错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17)在定时器操作过程中改变比较寄存器的相关注意事项。	p.96 □
			外部事件计数器	当读取外部事件计数器的计数值时，TM00 即为其值。	p.101 □
			脉冲宽度测量	如要使用两个捕捉寄存器，设置 TI000 和 TI010 引脚。	p.102 □
			方波输出	如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17)在定时器操作过程中改变比较寄存器的相关注意事项。	p.110 □
			PPG 输出	如在 TM00 操作中要改变 CR000 的设置可能引起错误。要改变设置，请参考 6.5 16 位定时器/事件计数器 00 (17)在定时器操作过程中改变比较寄存器的相关注意事项。	p.112 □
				以下为 CR000 和 CR010 的取值范围： 0000H < CR010 < CR000 ≤ FFFFH	p.113 □
				由 PPG 输出(CR00n 设置值+ 1)产生的脉冲周期的占空比为 (CR01n 设置值 + 1)/(CR00n 设置值 + 1)。	p.113 □
			单脉冲输出：软件触发	当单脉冲输出时，不要设置 OSPT00 位。在当前单脉冲输出完成后才可再次输出单脉冲。	p.115 □
	硬件			使用软件触发进行 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 TI000 及其复用端口引脚的电平。由于在这种情况下外部触发也有效，所以可使用 TI000 或其复用端口引脚对定时器清零&启动。这样会输出非预期的时序。	p.115 □
	软件			CR000 和 CR010 寄存器不能置 0000H。	p.116 □
				一旦为 TMC003 和 TMC002 位赋值（00 除外，操作停止模式），16 位定时器计数器 00 开始计数。	p.117 □

(6/18)

章节	类别	功能	功能详解	注意事项	页号
第六章	硬件	16 位定时器/事件计数器 00	单脉冲输出：外部触发	当单脉冲正在输出时不要再次再输入外部触发。在当前单脉冲输出完成后才可再次输出单脉冲。	p. 117 □
	软件			CR000 和 CR010 寄存器不能置 0000H 。	p. 118 □
				一旦为 TMC003 和 TMC002 位赋值（00 除外，操作停止模式）， 16 位定时器计数器 00 开始计数。	p. 119 □
	硬件		定时器启动误差	定时器启动后在信号匹配产生之前可能发生将近一个时钟的误差，这是由于 16 位定时器计数器 00 (TM00) 的启动与计数时钟不同步。	p. 120 □
	软件		单脉冲输出	一般只有在自由运行模式或清零&启动模式下，TI000 引脚有效沿输入才能有单脉冲输出。因为在清零&启动模式下， TM00 和 CR000 相等时不发生溢出，所以不可能输出单脉冲。	p. 121 □
			捕捉操作	如果 TI000 引脚的上升沿和下降沿同时作为输入有效沿，则捕捉操作无效。	p. 123 □
				当 CRC001 值为 1，如果 TI010 引脚有效沿被检测， TM00 的计数值不能被捕捉到 CR000 寄存器中，但是 TI010 引脚的输入可以用作外部中断源，因为可由外部有效信号输入产生 INTTM000 中断。	p. 123 □
			在定时器运行期间改变比较寄存器	16 位定时器捕捉/比较寄存器 0n0（CR0n0）作为比较寄存器使用，在定时器计数期间，在 16 位定时器计数器 00（TM00）和 16 位定时器捕捉/比较寄存器 0n0（CR0n0）的值相等的时序附近修改 CR0n0 的值时，改变 CR0n0 时序和相等时序可能会冲突，这种情况下，操作不能得到保证。在定时器计数时改变 CR0n0 的值时，INTTM000 中断服务程序有下面操作。	p. 124 □
				如果在定时器计数未执行上面的过程<1>期间改变 CR010 时，那么 CR010 中的值将被重新写入两次或更多，每次写入将引起 TO00 引脚电平的反转。	p. 124 □
第七章	软件	8 位定时器 80	CR80：比较寄存器 80	在定时器操作期间，不能改变 CR80 的值。如果在定时器操作期间改变 CR80 的值，则立刻产生中断请求信号。	p. 128 □
			TMC80：8 位定时器模式控制 80	定时器操作停止后必须对 TMC80 重新设置。	p. 129 □
				第 0 位和第 6 位必须清零。	p. 129 □
			间隔定时器	在定时器工作期间，不能改变 CR80 的值。如果在定时器工作期间改变 CR80 的值，则立刻产生相等中断请求信号。	p. 130 □
				当使用 8 位存储操作指令设置 TMC80 的计数时钟并同时允许 TM80 时，则在定时器启动后一个周期内的误差可能是一个或多个时钟。因此，要将 TM80 作为间隔定时器使用，必须按上述顺序进行设置。	p. 130 □

(7/18)

章节	类别	功能	功能详解	注意事项	页号
第七章	硬件	8 位定时器 80	定时器启动误差	从启动定时器到产生中断信号的时间包含了一个将近 1.5 个时钟周期的误差。这是因为如果在计数时钟为高电平时启动定时器，那么上升沿可能被立即检测到，并且计数器累加（参看**图 7－6**）。	p. 132 □
	软件		CR80: 比较寄存器 80	8 位比较寄存器 80（CR80）可设置为 00H。	p. 132 □
			STOP 模式	在执行 STOP 指令之前必须先停止定时器操作（TCE80=0）。	p. 132 □
第八章	软件	8 位定时器 H1	CMP01: 8 位定时 H 比较寄存器 01	在定时器计数期间不能修改 CMP01。	p. 135 □
			CMP11: 8 位定时 H 比较寄存器 11	在 PWM 输出模式中，当定时器停止记数后（TMHE1=0），再次启动记数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使前后设置的值相同也必须对 CMP11 重新设置）。	p. 135 □
			TMHMD1: 8 位 定时器 H 模式寄存器 1	当 TMHE1 设置为 1 时，禁止对 TMHMD1 寄存器的其它位进行设置。	p. 137 □
				在 PWM 输出模式下，当定时器停止记数后（TMHE1=0），再次启动记数操作时(TMHE1=1)，必须先对 CMP11 进行赋值（即使前后设置的值相同也必须对 CMP11 重新设置）。	p. 137 □
	硬件		PWM 输出	在 PWM 输出模式下，在重写寄存器之后，需要 3 个操作周期（使用 TMHMD1 寄存器的 CKS12 到 CKS10 位选择的信号）来传送 CMP11 寄存器的值。	p. 143 □
	软件			当定时器停止计数后（TMHE1=0），再次启动计数操作时（TMHE1=1），必须对 CMP11 赋值（即使设置的是相同的值，也必须再次赋值）。	p. 143 □
				CMP11 的取值（M）和 CMP01 的取值(N)应在如下范围内。 00H ≤ CMP11 (M) < CMP01 (N) ≤ FFH	p. 144 □
第九章	软件	看门狗定时器	WDTM: 看门狗定时器模式寄存器	第 7, 6, 5 位分别设置为 0, 1 和 1 (当通过可选字节选择“不能停止低速 Ring-OSC 时钟振荡器”，忽略其它值)。	p. 152 □
				复位后，仅能由 8 位存储器操作指令对 WDTM 写入一次。如果试图写入第二次，就会产生内部复位信号。	p. 152 □
				不能用 1 位存储器操作指令对 WDTM 进行操作。	p. 152 □
				当使用自写入方式对 flash 存储器进行操作时， 为看门狗设置足够的溢出时间 (例如 一字节写入: 最小 200 μs，一个块删除: 最小 10 ms)。	p. 152 □
			WDTE: 看门狗定时器使能寄存器	如果一个不是 ACH 的值被写入 WDTE，将产生一个内部复位信号。	p. 152 □
				如果使用 1 位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。	p. 152 □
				从 WDTE 读取的值是 9AH (与写入值 (ACH) 不同)。	p. 152 □

(8/18)

章节	类别	功能	功能详解	注意事项	页号
第九章	硬件	看门狗定时器	可选字节设置“低速内部振荡器不能被停止”	在这种模式中， 即使是执行 STOP 指令， 也不能停止看门狗定时器的操作。对于 8 位定时器 H1 (TMH1)，可选择低速内部时钟的一个分频作为计数源，因此在 STOP 指令执行后， 看门狗定时器溢出之前， 可使用 TMH1 的中断请求对看门狗定时器清零。如果没有执行该过程，则在 STOP 指令执行后当看门狗定时器溢出时会产生内部复位信号。	p. 153 □
			可选字节设置“低速内部振荡器能被停止”	此模式中，在 HALT/STOP 指令执行期间看门狗定时器的操作被停止。 在释放 HALT/STOP 模式后，通过使用在 HALT/STOP 指令执行前 WDTM 设置的看门狗定时器操作时钟将使计数器再次启动。此时，计数器并没有被清零，而是保持原值。	p. 155 □
第十章	软件	A/D 转换器	采样时间和 A/D 转换时间	以上采样时间和转换时间没有包括时钟频率误差。选择采样时间和转换时间时要将时钟频率误差考虑在内。	p. 161 □
			ADM：A/D 转换模式寄存器	以上采样时间和转换时间没有包括时钟频率误差。选择转换时间时还要将时钟频率误差考虑在内。	p. 166 □
				当 A/D 转换结束(ADCS = 0)时，对 ADM 中除了 ADCS 的位进行操作后，要执行两条 NOP 指令或一条相当于 2 个机器周期的指令，再设置 ADCS=1，A/D 转换重新开始。	p. 166 □
				在重写 FR0 至 FR2 时，A/D 转换必须停止 (ADCS = 0)。	p. 166 □
				一定要把第 6 位、第 2 位和第 1 位清 0。	p. 166 □
			ADS：模拟输入通道指定寄存器	ADS 的第 2 位到第 7 位必须清 0。	p. 167 □
			ADCR：10 位 A/D 转换结果寄存器	当对 A/D 转换模式寄存器（ADM）和模拟输入通道选择寄存器（ADS）进行写操作时，ADCR 的内容可能不确定。在对 ADM 和 ADS 执行写操作之前，要在转换完成后读取转换结果。在其他时间读取可能读到错误的转换结果。	p. 167 □
			PMC2：端口模式控制寄存器 2	当 PMC20 至 PMC23 被设置为 1 时，P20/ANI0 至 P23/ANI3 引脚不能被用作端口引脚。	p. 168 □
			A/D 转换器操作	必须确保步骤〈1〉到〈4〉的操作时间为 1 μs 或更长。	p. 173 □
				若将步骤〈1〉和〈2〉次序颠倒，也不会有问题。	p. 173 □
				〈1〉 可以被忽略。 但在这种情况下中当步骤〈4〉执行后忽略第一个转换结果。	p. 173 □
				步骤〈5〉到〈8〉所经历的时间与 ADM 的第 5 位到第 3 位(FR2～FR0)设置的转换时间不同。步骤〈7〉到〈8〉所经历的时间即为转换时间，由 FR2 到 FR0 设置。	p. 173 □

(9/18)

章节	类别	功能	功能详解	注意事项	页号
第十章	硬件	A/D 转换器	STOP 模式下的工作电流	在 STOP 模式下，A/D 转换停止操作。此时，将 A/D 转换模式寄存器（ADM）的位 7（ADCS）和位 0（ADCE）清 0，可以降低操作电流。	p. 176 □
			ANI0 到 ANI3 的输入范围	观察 ANI0 至 ANI3 的输入电压的额定范围。如果输入到模拟输入通道的电压为 V_{DD} 或更高，或者为 V_{SS} 或更低（即使在额定的最大范围内），则该通道的转换值是不确定的。此外，其他通道的转换值也可能受影响。	p. 176 □
	软件		运行冲突	转换结束后，指令对 A/D 转换结果寄存器（ADCR，ADCRH）的写操作和 ADCR，ADCRH 的读操作同时进行，就为冲突操作。 ADCR,ADCRH 读操作优先。在读操作执行完后，新的转换结果将写入到 ADCR、ADCRH。	p. 176 □
				转换结束后，对 ADCR,ADCRH 写操作和 A/D 转换模式寄存器(ADM)写操作，或者对模拟输入通道选择寄存器(ADS)写操作同时进行，就为冲突操作。 转换结束后，对 ADCR,ADCRH 写操作和 A/D 转换模式寄存器(ADM)写操作，或者对模拟输入通道选择寄存器(ADS)写操作同时进行，就为冲突操作。	p. 176 □
	硬件		噪声对策	为了保持 10 位分辨率，必须注意输入到 VDD 引脚和 ANI0 至 ANI3 引脚的噪声。 <1>在低阻抗和高频反馈的电源上连接一个电容。 <2>由于噪声的影响和模拟输入源的输出阻抗成正比，因此建议外接一个电容，如图 9-19 所示，以降低噪声。 <3>在转换期间，不要将引脚 ANI0 到 ANI3 的 A/D 转换功能切换到它们的复用功能。 <4>在转换开始后，立即设置 HALT 模式，可以提高转换精度。	p. 176 □
			ANI0/P20 到 ANI3/P23	模拟输入引脚(ANI0 至 ANI3) 也可用作输入端口引脚 (P20 至 P23)。 当选择 ANI0 至 ANI3 进行 A/D 转换时，在转换过程中不要访问 P20 至 P23，否则会降低转换效果。	p. 177 □
				如果正在进行 A/D 转换的引脚相邻引脚有数字脉冲，则由于噪声耦合，有可能得不到预期的 A/D 转换值。因此，在进行 A/D 转换时不要在相邻引脚输入脉冲信号。	p. 177 □
			ANI0 到 ANI3 引脚的输入阻抗	在 A/D 转换器中，采样期间内内部采样电容充电，采样时间接近转换时间的 1/6。 由于在采样期间有漏电流和电容充电的电流，因此输入阻抗在采样期间或其他时候都有波动。 如果在使用参考电压的最短转换时间，要进行充足的采样。建议模拟输入源的输出阻抗为 1kΩ或更小，或者在引脚连接 0.01μF 至 0.1μF 的电容(见 图 10-19)。	p. 177 □

(10/18)

章节	类别	功能	功能详解	注意事项	页号
第十章	软件	A/D 转换器	ADIF 中断请求标志 (ADIF)	即使模拟输入通道选择寄存器（ADS）的值被修改，中断请求信号 (ADIF)也不能清零。 因此，如果在 A/D 转换期间有模拟输入引脚改变，那么在重写 ADS 前，A/D 转换的结果和模拟输入改变前的 ADIF 可能被设置。注意在 ADS 重写后立即读取 ADIF 时，尽管模拟输入改变后的 A/D 转换并未结束，也会设置 ADIF。 在 A/D 转换结束要重新开始转换时，要在重新开始转换前先将 ADIF 清 0。	p.177 □
			A/D 转换开始后的转换结果	A/D 转换开始后，如果在 ADCE 置 1 后 1 μs 内对 ADCS 置 1，或在 ADCE 为零时对 ADCS 置 1，那么第一次 A/D 转换的值可能不在额定范围内。可采取措施，如将 A/D 转换结束中断请求（INTAD）挂起，并删除第 1 次转换结果。	p.178 □
			A/D 转换结果寄存器 (ADCR, ADCRH) 读操作	当对 A/D 转换器模式寄存器（ADM）和模拟输入通道选择寄存器（ADS）进行读取时，ADCR 和 ADCRH 的内容可能变得不确定。要在 ADM 和 ADS 进行写操作前并且转换结束后，读取转换结果。若采用和上述不同的时序，可能会读取到不正确的转换结果。	p.178 □
第十一章	硬件	串行接口 UART6	UART 模式	TxD6 输出反向功能的操作仅在发送端进行而不在接收端进行。如要使用这个功能，接收端应准备好接收已反向的数据。	p.179 □
	软件			如果串行接口 UART6 使用的时钟未停止（例如，处于 HALT 模式下），则正常操作继续。如果时钟已停止（例如，处于 STOP 模式下），则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。TxD6 引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使 POWER6 = 0, RXE6 = 0, TXE6 = 0。	p.179 □
				如果连续传送数据，从停止位到下一个起始位的通讯时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果在 LIN 模式中，则不能使用连续发送功能。	p.179 □
			TXB6：发送缓冲寄存器 6	当异步串行接口发送状态寄存器 6 (ASIF6)的第 1 位(TXBF6)为 1 时不要对 TXB6 进行写操作。	p.185 □
				在通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），不要通过软件刷新（写入相同值）TXB6。	p.185 □

(11/18)

章节	类别	功能	功能详解	注意事项	页号
第十一章	软件	串行接口 UART6	ASIM6：异步串型接口操作模式寄存器 6	启动操作时，对 POWER6 置 1，然后对 TXE6 置 1。要停止操作，先将 TXE6 清零，然后将 POWER6 清零。	p.187 □
				启动操作时，对 POWER6 置 1，然后对 RXE6 置 1。要停止操作，先将 RXE6 清零，然后将 POWER6 清零。	p.187 □
				当 RxD6 引脚输入为高电平时，先对 POWER6 置 1，再对 RXE6 置 1。当输入为低电平 POWER6 置 1， RXE6 置 1 时，启动接收操作。	p.187 □
				在对 PS61、PS60 和 CL6 位重写之前，先将 TXE6 和 RXE6 位清零。	p.187 □
				在启动 LIN 设备时 PS61 与 PS60 恒为 0。	p.187 □
				在重写 SL6 位时必须确保 TXE6 = 0。当“停止位的个数=1”时，始终执行接收操作，因此接收不会受到 SL6 的影响。	p.187 □
				当重写 ISRM6 时，RXE6 必须为 0。	p.187 □
			ASIS6：异步串型接口接收错误状态寄存器 6	根据异步串行接口操作模式寄存器 6(ASIM6)的 PS61 和 PS60 位的内容的不同，对 PE6 位的操作也有所不同。	p.188 □
				无论停止位的个数是多少，只能将接收数据的第 1 位作为停止位进行校验。	p.188 □
				如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 6（RXB6）中而是被忽略。	p.188 □
			ASIF6：异步串型接口发送状态寄存器 6	如要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 中。检查 TXBF6 标志位，必须为“0”。然后，将下一个发送数据（第二个字节）写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。	p.189 □
				若要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。	p.189 □
			CKSR6：时钟选择寄存器 6	当重写 TPS63 ～ TPS60 时，必须确保 POWER6 = 0。	p.190 □
			BRGC6：波特率发生控制寄存器 6	在重写 MDL67 ～ MDL60 位时，必须确保 ASIM6 的第 6 位(TXE6)和第 5 位(RXE6) = 0。	p.191 □
	硬件			波特率为 8 位计数器输出时钟的 1/2。	p.191 □
	软件		ASICL6：异步串型接口控制寄存器 6	通信过程中（当异步串行接口操作模式寄存器 6（ASIM6）的第 7 位（POWER6）和第 6 位（TXE6）均为 1，或 ASIM6 的第 7 位（POWER6）和第 5 位（RXE6）均为 1），可由软件刷新（写入相同值）ASICL6。 但是，在 SBF 接收(SBRF6 = 1)或 SBF 发送(在 SBTT6 设置 (1) 与 INTST6 产生之间) 期间，执行刷新操作设置 SBRT6 = 1 且 SBTT = 1，将会再次触发 SBF 接收和 SBF 发送，因此不要设置 SBRT6 = 1 与 SBTT = 1。	p.192 □
				如果出现 SBF 接收错误，则再次返回到 SBF 接收模式下，并保持 SBRF6 标志位不变。	p.193 □

(12/18)

章节	类别	功能	功能详解	注意事项	页号
第十一章	软件	串行接口 UART6	ASICL6：异步串行接口控制寄存器 6	在设置 SBRT6 为 1 之前，必须确保 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)为 1。并且，在设置 SBRT6 为 1 之后，不要在 SBF 接收结束(产生中断请求信号)之前将 SBRT6 清零。	p.193 □
				SBRT6 的读取值恒为 0。在 SBF 正确接收后 SBRT6 自动清零。	p.193 □
				在设置 SBTT6 为 1 之前，必须确保 ASIM6 的第 7 位(POWER6)和第 6 位(TXE6)为 1。并且，在设置 SBTT6 为 1 之后，不要在 SBF 发送结束(产生中断请求信号)之前将 SBTT6 清零。	p.193 □
				SBTT6 的读取值恒为 0。在 SBF 发送结束时 SBRT6 自动清零。	p.193 □
				在重写 DIR6 和 TXDLV6 之前，将 TXE6 和 RXE6 清零。	p.193 □
			POWER6，TXE6，和 RXE6：ASIM6 的第 7，6，和 5 位	TXE6 和 RXE6 清零后再对 POWER6 清零，可设置操作停止模式。 若要启动操作，设置 POWER6 为 1，然后设置 TXE6 为 1 且 RXE6 为 1。	p.195 □
			UART 模式	在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。	p.196 □
			奇偶校验和操作	在 LIN 模式中，PS61 与 PS60 恒为 0。	p.200 □
			连续发送	在连续发送期间，ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10” 变成“11”、再变成“01”。因此检查状态时，不能通过比较 TXBF6 和 TXSF6 标志来判断。只能根据读取 TXBF6 标志的值来判断连续发送是否可能。	p.202 □
				如果是在 LIN 模式下，则不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6(TXB6) 之前，异步串行接口发送状态寄存器(ASIF6) 的值必须为 00H。	p.202 □
			连续发送期间的 TXBF6：ASIF6 的第 1 位	若要连续发送数据，将第一个发送的数据（第一个字节）写入 TXB6 中。检查 TXBF6 标志位，必须为“0”。然后，将下一个发送数据（第二个字节）写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。	p.202 □
			连续发送期间的 TXSF6：ASIF6 的第 0 位	若要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。	p.202 □
				在连续发送期间可能会产生错误，即在发送完一帧数据后 INTST6 中断服务执行之前下一个发送操作已完成。可通过编写一段计算发送数据的个数和检查 TXSF6 标志的程序来检测错误。	p.202 □

(13/18)

章节	类别	功能	功能详解	注意事项	页号
第十一章	软件	串行接口 UART6	正常接收	即使出现接收错误，也必须读取接收缓冲寄存器 6 (RXB6)的内容。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。	p. 206 □
				接收始终按“停止位的个数为 1”的情况执行，第 2 个停止位被忽略。	p. 206 □
				在读取 RXB6 之前，必须先读取异步串行接口接收错误状态寄存器 6(ASIS6)。	p. 206 □
			串行时钟的发生	在接收端保持发送期间的波特率误差在一个允许的误差范围内。	p. 212 □
				接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。	p. 212 □
			接收期间允许的波特率的范围	必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。	p. 214 □
第十二章	软件	乘法器	MUL0：16 位乘法结果存储寄存器 0	这个寄存器既能用 16 位存储器操作指令操作，也能用 8 位存储器操作指令操作，使用 8 位存储器操作指令对其进行操作时，须使用直接寻址方式对其进行访问。	p. 217 □
			MULC0：控制寄存器 0	一定要将第 1 位到第 7 位清零。	p. 219 □
第十三章	硬	中断功能	向量表地址	向量表地址 0014H 没有相对应的中断源。	p. 222 □
	软件		IF0, IF1：中断请求标志寄存器, MK0, MK1：中断屏蔽标志寄存器	由于 P30, P31, P41, 和 P43 有一个复用功能就是作为外部中断输入，所以当指定端口输出模式使输出电平发生变化，中断请求标志置 1。因此，在使用输出模式之前应先将中断屏蔽标志置 1。	pp. 225, 226 □
			INTM0：外部中断模式寄存器 0	第 1 位和第 0 位必须清零。	p. 227 □
				在设置 INTM0 寄存器前，必须设置相应的中断屏蔽标志 (××MK× = 1) 以禁止中断。INTM0 寄存器设置完成后，首先对中断请求标志(××IF× = 0)进行清零，然后再对中断屏蔽标志 (××MK× = 0)进行清零以允许中断。	p. 227 □
			INTM1：外部中断模式寄存器 1	第 2 位至第 7 位必须清零。	p. 228 □
				设置 INTM1 之前，先设定 PMK3 为 1 以禁止中断。然后对 PIF3 清 0 再对 PMK3 清 0 允许中断。	p. 228 □
			中断请求不确定	当中断请求标志寄存器（IF0,IF1）或中断屏蔽标志寄存器（MK0,MK1）被访问期间，中断请求等待。	p. 231 □

(14/18)

章节	类别	功能	功能详解	注意事项	页号
第十四章	软件	待机功能	–	仅当由可选字节选择低速 Ring-OSC 振荡器“可通过软件停止”时，LSRSTOP 设置才有效。	p.234 □
	硬件		STOP 模式	当改变到 STOP 模式时，一定要在执行 STOP 指令(除了以低速 Ring-OSC 时钟工作的外部硬件之外)之前停止外部硬件操作。	p.235 □
			STOP 模式 HALT 模式	当使用待机功能时推荐按以下步骤减少 A/D 转换器的工作电流：首先对 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)和第 0 位 (ADCE)清零以停止 A/D 转换，然后执行 HALT 或 STOP 指令。	p.235 □
			STOP 模式	如果在置 STOP 模式之前低速 Ring-OSC 振荡器正在工作，则在 STOP 模式中不能停止低速 Ring-OSC 时钟 (参见 表 14-1)。	p.235 □
	软件		OSTS：振荡稳定时间选择寄存器	如要设置并释放 STOP 模式，按以下方式设置振荡稳定时间。 预期振荡稳定时间 ≤ OSTS 设置的振荡稳定时间	p.236 □
	硬件			无论是复位后还是由中断产生释放 STOP 模式，释放 STOP 模式后的等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间 (即下图“a” 所示的部分)。	p.236 □
	软件			由可选字节选择上电或复位释放后经过的振荡稳定时间。详情可参见第十八章可选字节。	p.236 □
			HALT 模式设置及操作	由于中断请求信号可用来释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即释放待机模式。	p.237 □
			STOP 模式设置及操作	由于中断请求信号用于释放待机模式，如果某一中断源中断请求标志置位且中断屏蔽标志清零，则立即退出待机模式。因此，在 STOP 模式中，执行 STOP 指令后恢复正常操作模式然后操作停止 34 μs (典型值) (在经过由振荡稳定时间选择寄存器(OSTS)设置的等待时间后，晶体/陶瓷振荡器启动)。	p.240 □
第十五章	硬件	复位功能	–	对于外部复位，输入 RESET 引脚的低电平的时间应为 2μs 或更长。	p.244 □
				复位输入期间，系统时钟和低速内部振荡时钟停止振荡。	p.244 □
				RESET 引脚只能被复用输入端口引脚(P34)，如果在由 POC 电路释放复位后且在再次参考选项字节之前低电平输入到 RESET 引脚，78K0S/KU1+ 和 78K0S/KY1+复位。复位状态保持直到高电平输入到 RESET 引脚。	p.244 □
				LVI 电路的内部复位信号不会使 LVI 电路复位。	p.245 □
			看门狗定时器溢出时的复位时序	在看门狗定时器内部复位的情况下，看门狗定时器也复位。	p.247 □

(15/18)

章节	类别	功能	功能详解	注意事项	页号
第十	软件	复位功能	RESF：复位控制标志寄存器	不能通过 1 位存储器操作指令读取数据。	p.251 □
第十六章	软件	上电清零电路	上电清零电路的功能	如果 POC 电路产生内部复位信号，则将复位标志寄存器（RESF）清零。	p.252 □
	硬件			POC 电路的检测电压(V_{POC})值范围为 2.1 V ±0.1 V，故正常工作电压范围为 2.2 ～5.5 V。	p.252 □
	软件		上电清零电路的注意事项	对于一个电源电压在某段时间内在 POC 检测电压(VPOC)附近波动的系统，可能会反复进行复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所经历的时间。	p.254 □
第十七章	软件	低电压检测	LVIM：低电压检测寄存器	可在执行完以下任一过程后终止 LVI •当使用 8 位存储器操作指令时：写 00H 到 LVIM。 •当使用 1 位存储器操作指令时：对 LVION 清零。	p.257 □
				第 2～6 位必须清零。	p.257 □
			LVIS：低电压检测电平选择寄存器	第 4～7 位必须清零。	p.258 □
			用于复位	必须执行过程<1>，当 LVIMK = 0 时，在过程<3>之后可能立即会产生中断。	p.259 □
				当 LVIM 被设置为 1 时，如果电源电压 (V_{DD}) ≥ 检测电压 (V_{LVI})，则不会产生内部复位信号。	p.259 □
			低电压检测注意事项	在一个电源电压(VDD)在某段时间内在 LVI 检测电压附近波动的系统中，以下操作决定于如何使用低电压检测器。 <1>用作复位 系统可能会反复进行复位和释放复位。 在这种情况下，可采用下面的方法(1)任意设置从复位释放到微控制器的启动所经历的时间。 <2>用作中断 可能会频繁产生中断请求。可采用方法(2)。	p.262 □
第十八章	硬件	可选字节	低速内部振荡	如果选择低速 Ring-OSC 时钟振荡不能被停止，看门狗定时器(WDT)的计数器时钟就会固定为低速 Ring-OSC 时钟。	p.266 □
				如果选择低速 Ring-OSC 时钟可以通过软件停止，不管低速 Ring-OSC 模式寄存器(LSRCM)的第 0 位(LSRSTOP)设置如何，WDT 的计数器时钟在 HALT/STOP 模式下被停止，。同样地，当选择除低速 Ring-OSC 时钟之外的任何一个时钟作为 WDT 的计数器时钟时，时钟提供也被停止。但是，如果选择低速 Ring-OSC 时钟作为 8 位定时器 H1 的计数时钟，那么在低速 Ring-OSC 工作时(LSRSTOP = 0)，在 STOP 模式下计数时钟可提供给 8 位定时器 H1 使用。	p.266 □

(16/18)

章节	类别	功能	功能详解	注意事项	页号
第十八章	硬件	可选字节	系统时钟源的选择	因为 X1 和 X2 引脚也作为 P121 和 P122 引脚使用， X1 和 X2 引脚使用的条件不同，取决于系统时钟源的选择。 (1) 选择晶体/陶瓷振荡时钟 X1 和 X2 引脚不能作为 I/O 端口使用，因为它们正作为时钟输入端口 (2) 选择外部时钟输入 因为 X1 引脚作为外部时钟输入口，P121 口不能作为 I/O 口使用 (3) 选择高速内部振荡器时钟 P121 和 P122 端口可作为 I/O 端口	p.266 □
			复位引脚的控制	在通过上电清零功能释放复位状态后并且在再次参考可选字节之前，一个低电平输入到 RESET 引脚，78K0S/KB1+复位，并且状态保持到一个高电平输入到 RESET 引脚。	p.266 □
			上电或复位释放后振荡稳定时间	仅当选晶体/陶瓷振荡时钟作为为系统时钟源时,这项设定才有效。 如果选用高速 Ring-OSC 时钟或外部时钟输入作为系统时钟源时,无需等待时间。	p.267 □
第十九章	软件	Flash 存储器	PG-FP4 GUI 软件设置值举例	以上值为推荐值。由于环境不同，这些值可能改变，所以请使用时请先估计一个值。	p.275 □
			安全设置	当下一次设定编程模式时，安全设置有效。	p.279 □
				在设定成批擦除的安全设置之后，擦除指令不能执行。另外，即使一个写入命令被执行，与已写入的数据不同的部分是不能被写入的，因为擦除命令已无效。	p.279 □
			自编程功能	在执行自写入以前，自写入处理程序必须先被包含在用户程序中。	p.280 □
				如果在自编程期间发生中断，中断请求标志置 1，在自编程模式结束后才能被响应。为了避免这种情况，应在自编程期间或者从普通模式切换到自编程模式之前以一定顺序终止中断服务(通过将 MK0 和 MK1 设定为 0FFH，并执行 DI 指令)。	p.283 □
				当自编程命令执行时，其它命令都不能执行。因此，要先清除和重启看门狗时钟计数器，以便在自编程时看门狗计数器不会溢出。对于执行自编程所需时间可参考表 19-11。	p.283 □
				当执行自编程命令时，不使用 RAM。	p.283 □
				当 FLASH 存储器正在写入或擦除时，如果供电电压下降或者有复位信号输入，则此次写入或擦除操作不一定成功。	p.283 □
				在块擦除期间空白数据的值设为 0FFH。	p.283 □

(17/18)

章节	类别	功能	功能详解	注意事项	页号
第十九章	软件	Flash 存储器	自编程功能	当选用振荡器或外部时钟做为主时钟时，从自编程模式的设定到 HALT 指令的执行需要 16μs 的等待时间。	p.283 □
				自编程模式下和 HALT 模式下引脚的状态是相同的。	p.283 □
				在自编程模式中，因为禁止通过 on-board/off-board 编程设置安全功能，所以不管安全功能如何设定如何，自编程命令都可执行。如要在自编程期间禁止写入或擦除，则需要设定保护字节。	p.283 □
				当执行自编程命令时，如果 FLASH 地址指针 H(FLAPH)和 FLASH 地址指针 H 比较寄存器(FLAPHC)的第 4 位到第 7 位为 1，在执行自编程命令之前一定要对这些位清零。	p.283 □
			FLPMC: Flash 编程模式控制寄存器	以下是自编程模式的注意事项： •在自编程期间发生的中断，在自编程模式结束后才能被响应。为了避免这种情况，应在从普通模式切换到自编程模式之前以一定顺序终止中断服务(通过将 MK0 和 MK1 设定为 0FFH，并执行 DI 指令)。 •当自编程命令执行时，其它命令都不能执行。因此，要先清除和重启看门狗时钟计数器，以便在自编程时计数器不会溢出。参考表 19-11 时钟对应于自编程的执行。 •当 FLASH 存储器正在数据写入或擦除过程中，如果供电电压下降或者有复位信号输入，则此次写入或擦除操作不一定成功。	p.284 □
				当选择振荡器或外部时钟作为主时钟时，从 FLSPM 被设置为 1 到 HALT 指令的执行需要 16 μs 的等待时间。	p.284 □
			PFCMD: Flash 保护命令寄存器	当指定顺序执行时，要禁止中断服务(可通过将 MK0 和 MK1 设为 0FFH，还有执行 DI 指令来实现)。	p.285 □
			FLAPH, FLAPL: Flash 地址指针 H 和 L	在自编程命令执行以前，确保 FLASH 地址指针 H(FLAPH)的第 4 位到第 7 位和 FLASH 地址指针 H 比较寄存器(FLAPHC)清零。	p.288 □
			FLAPHC, FLAPLC: Flash 地址指针 H/L 比较寄存器	在自编程命令执行以前，确保 FLASH 地址指针 H(FLAPH)的第 4 位到 7 位和 FLASH 地址指针 H 比较寄存器(FLAPHC)清零。	p.288 □
				当进行块擦除、校验、空白检测时，将块的序号赋值给 FLAPHC(与 FLAPH 的值相同)。	p.288 □
				当块擦除完成时 FLAPLC 要清零，当块空白检测完成后 FLAPLC 要设为 0FFH。	p.288 □

(18/18)

<table>
<tr><th>章节</th><th>类别</th><th>功能</th><th>功能详解</th><th>注意事项</th><th>页号</th></tr>
<tr><td rowspan="3">第十九章</td><td rowspan="3">软件</td><td rowspan="3">存储器</td><td>切换到自编程模式</td><td rowspan="2">在执行上述操作时，必须保证用户程序所使用的地址没有数据擦除和写入操作发生。</td><td rowspan="2">pp. 291, 292, 294, 295 □</td></tr>
<tr><td>切换到正常模式</td></tr>
<tr><td>字节写入</td><td>如果写入失败，应先擦除块内容并再次写入。</td><td>p. 303 □</td></tr>
<tr><td rowspan="2">第二十一章</td><td rowspan="2">硬件</td><td rowspan="2">电气特性</td><td>最大额定值</td><td>任何一项参数哪怕是在瞬间超过最大额定值,都会使产品质量受到影响. 也就是说， 最大额定值是产品濒临物理损坏的临界点， 因此,必须保证产品在不超过最大额定值的条件下使用。</td><td>p. 337 □</td></tr>
<tr><td>X1 振荡器特性</td><td>在使用 X1 振荡器时， 上表中的虚线框内部分的连线应遵从如下的连接方式，以防止线间电容产生不利影响。

• 连接线越短越好。
• 连接线不与其他信号线交叉。
• 如果信号线流经的电流变化较大,则不要在其周围连线。
• 要保持振荡器电容器的接地点电压与 VSS 相同。
• 避免大电流从电容到地的连线上流过。
• 不要从振荡器获取信号。</td><td>p. 338 □</td></tr>
</table>